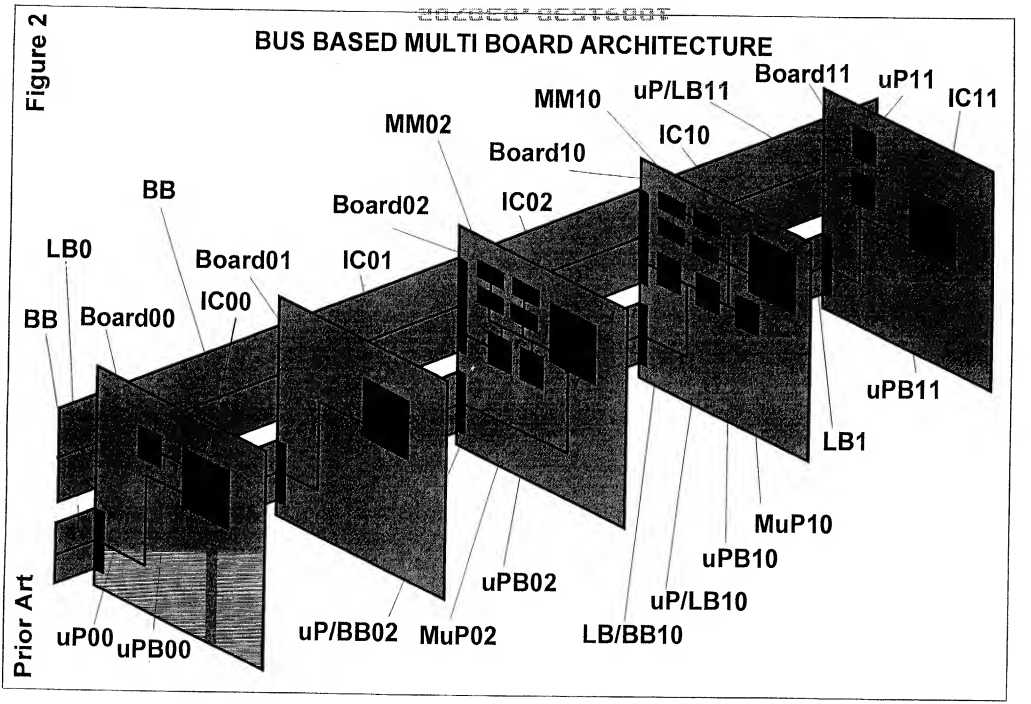
Prior Art

Figure 1

Figure 2


BUS BASED MULTI BOARD ARCHITECTURE
Board11
uP11
IC11
MM10
uP/LB11
IC10
MM02
Board10
BB
Board02
IC02
LB0
Board01
IC01
IC00
BB
Board00
uPB11
LB1
MuP10
uPB10
uPB02
uP/LB10
uP/BB02
MuP02
LB/BB10
uP00
uPB00
Prior Art

Prior Art

Figure 3

IC WITH uP INTERFACE AND LOCAL BUS INTERFACE

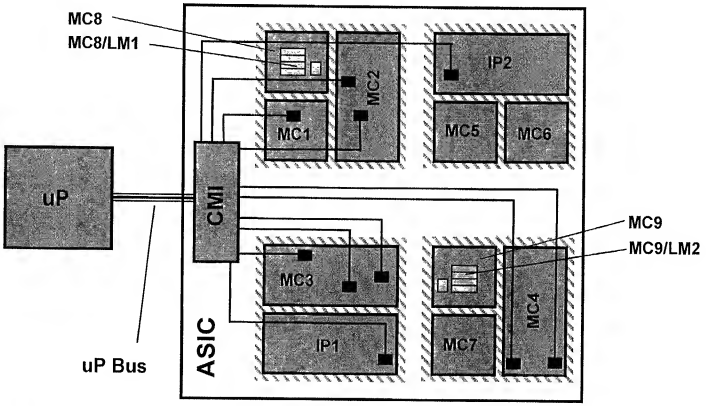
Figure 4

Prior Art

IC WITH LOCAL BUS INTERFACE

Figure 5

ASIC implementation of CMI with CMPI macro-cells interface



Prior Art

Figure 6

Prior Art

ASIC implementation of CMI with CBBI macro-cells interface

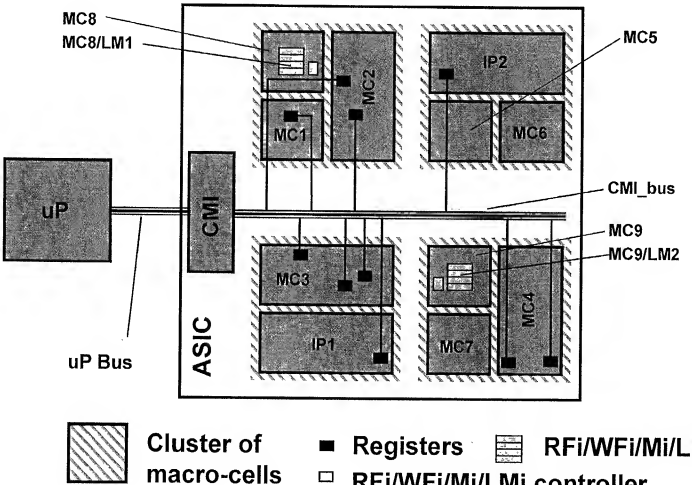


Figure 7

Prior Art

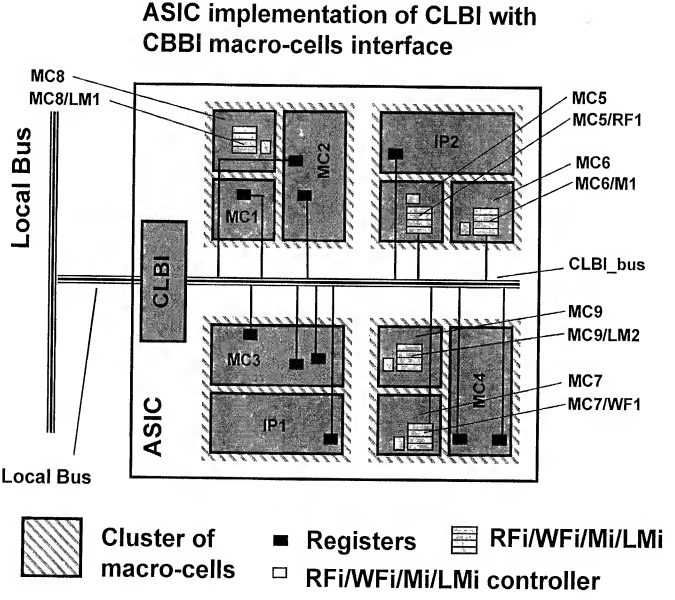


Figure 8

Prior Art

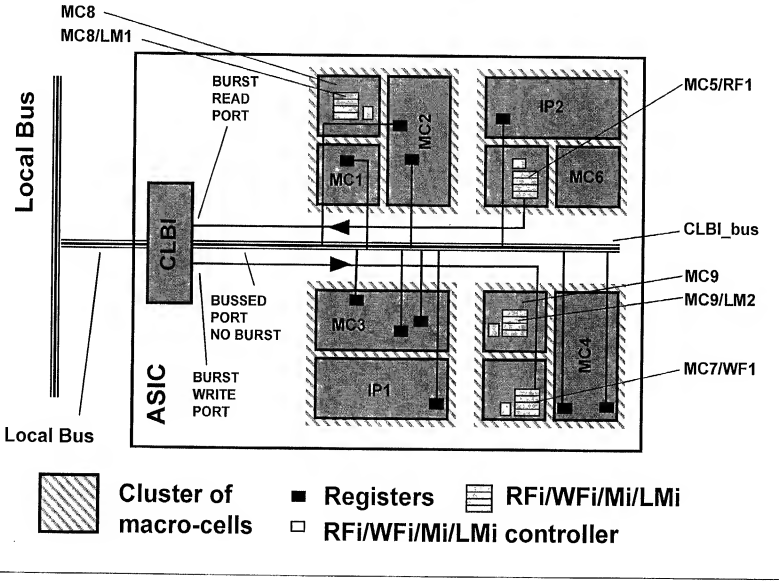


Figure 9

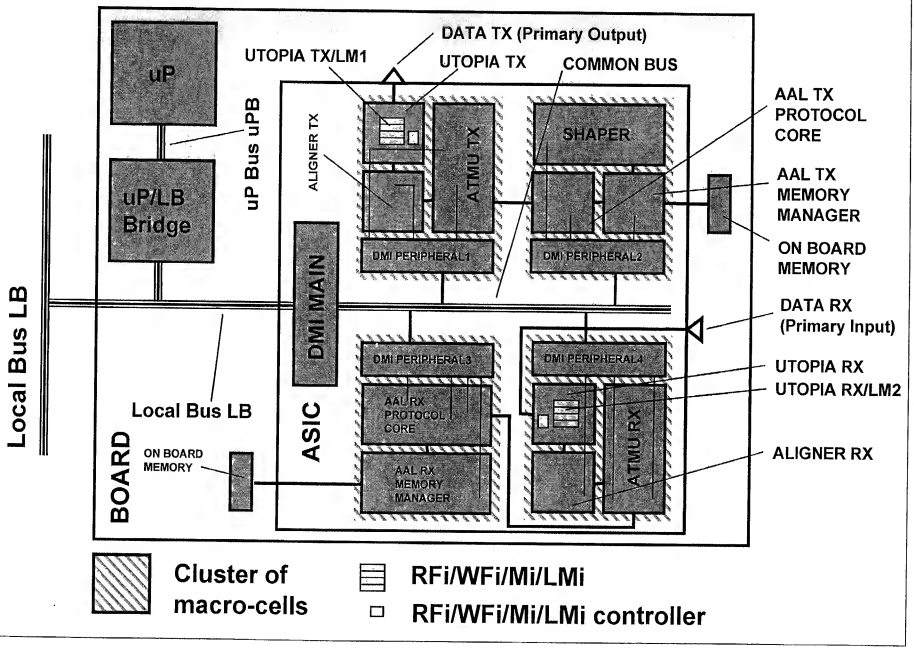

Board hosting ASIC implementation of AAL5 interfaced via DMI
uP
uP/LB Bridge
uP Bus uPB
Local Bus LB
Local Bus LB
BOARD
ON BOARD MEMORY
ASIC
DMI MAIN
ALIGNER TX
UTOPIA TX/LM1
DATA TX (Primary Output)
UTOPIA TX
COMMON BUS
ATMU TX
SHAPER
DMI PERIPHERAL1
DMI PERIPHERAL2
DMI PERIPHERAL3
DMI PERIPHERAL4
AAL RX PROTOCOL CORE
AAL RX MEMORY MANAGER
ATMU RX
AAL TX PROTOCOL CORE
AAL TX MEMORY MANAGER
ON BOARD MEMORY
DATA RX (Primary Input)
UTOPIA RX
UTOPIA RX/LM2
ALIGNER RX
Cluster of macro-cells
RFi/WFi/Mi/LMi
RFi/WFi/Mi/LMi controller

Figure 10

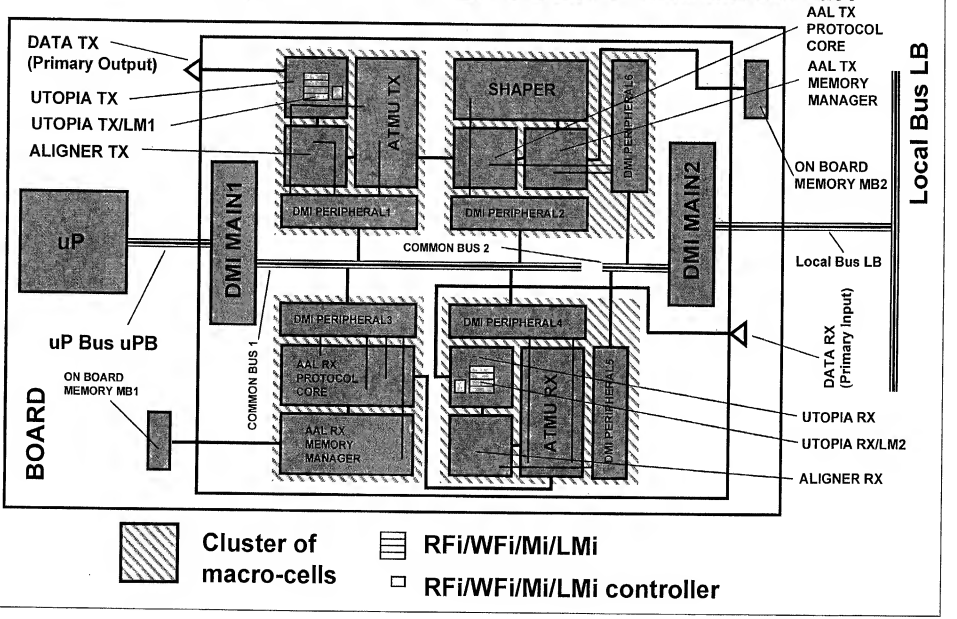

Board hosting FPGA bread-boarding implementation
of a DMI for microprocessor interface and a DMI for local bus interface
DATA TX
(Primary Output)
UTOPIA TX
UTOPIA TX/LM1
ALIGNER TX
uP
uP Bus uPB
ON BOARD
MEMORY MB1
BOARD
DMI MAIN1
COMMON BUS 1
ATMU TX
SHAPER
DMI PERIPHERAL1
DMI PERIPHERAL2
DMI PERIPHERAL6
COMMON BUS 2
DMI MAIN2
DMI PERIPHERAL3
DMI PERIPHERAL4
AAL RX
PROTOCOL
CORE
AAL RX
MEMORY
MANAGER
ATMU RX
DMI PERIPHERAL5
AAL TX
PROTOCOL
CORE
AAL TX
MEMORY
MANAGER
ON BOARD
MEMORY MB2
Local Bus LB
Local Bus LB
DATA RX
(Primary Input)
UTOPIA RX
UTOPIA RX/LM2
ALIGNER RX
Cluster of
macro-cells
RFi/WFi/Mi/LMi
RFi/WFi/Mi/LMi controller

Figure 11

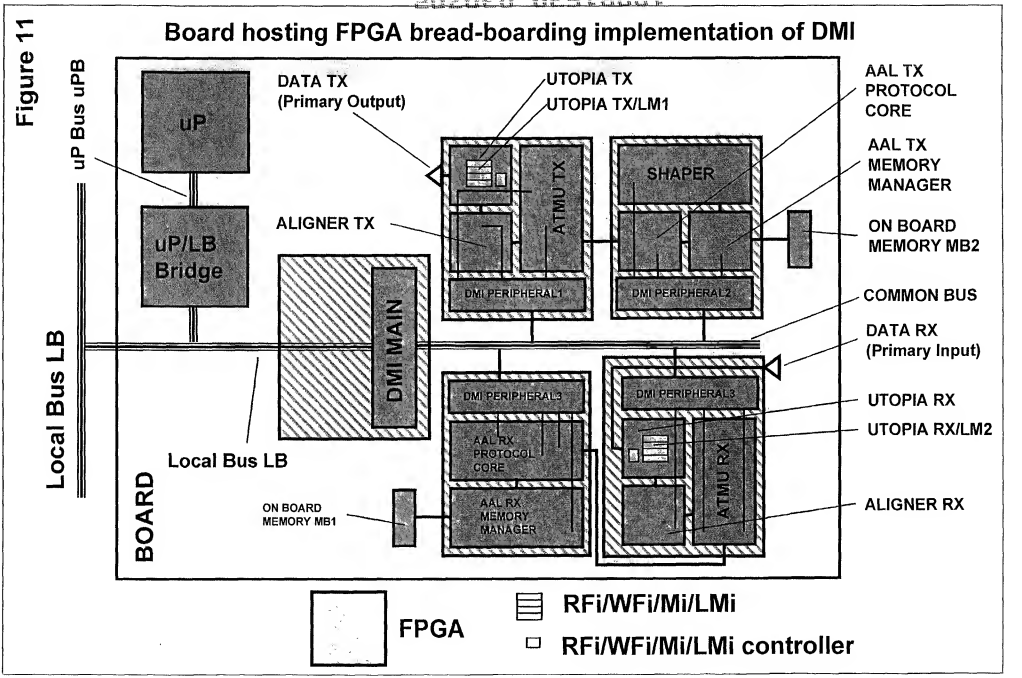

Board hosting FPGA bread-boarding implementation of DMI
uP
uP/LB Bridge
uP Bus uPB
Local Bus LB
BOARD
DATA TX
(Primary Output)
UTOPIA TX
UTOPIA TX/LM1
ATMU TX
SHAPER
AAL TX
PROTOCOL
CORE
AAL TX
MEMORY
MANAGER
ALIGNER TX
ON BOARD
MEMORY MB2
DMI PERIPHERAL1
DMI PERIPHERAL2
DMI MAIN
COMMON BUS
DATA RX
(Primary Input)
DMI PERIPHERAL3
DMI PERIPHERAL3
UTOPIA RX
UTOPIA RX/LM2
AAL RX
PROTOCOL
CORE
ATMU RX
Local Bus LB
ON BOARD
MEMORY MB1
AAL RX
MEMORY
MANAGER
ALIGNER RX
FPGA
RFi/WFi/Mi/LMi
RFi/WFi/Mi/LMi controller

Figure 12

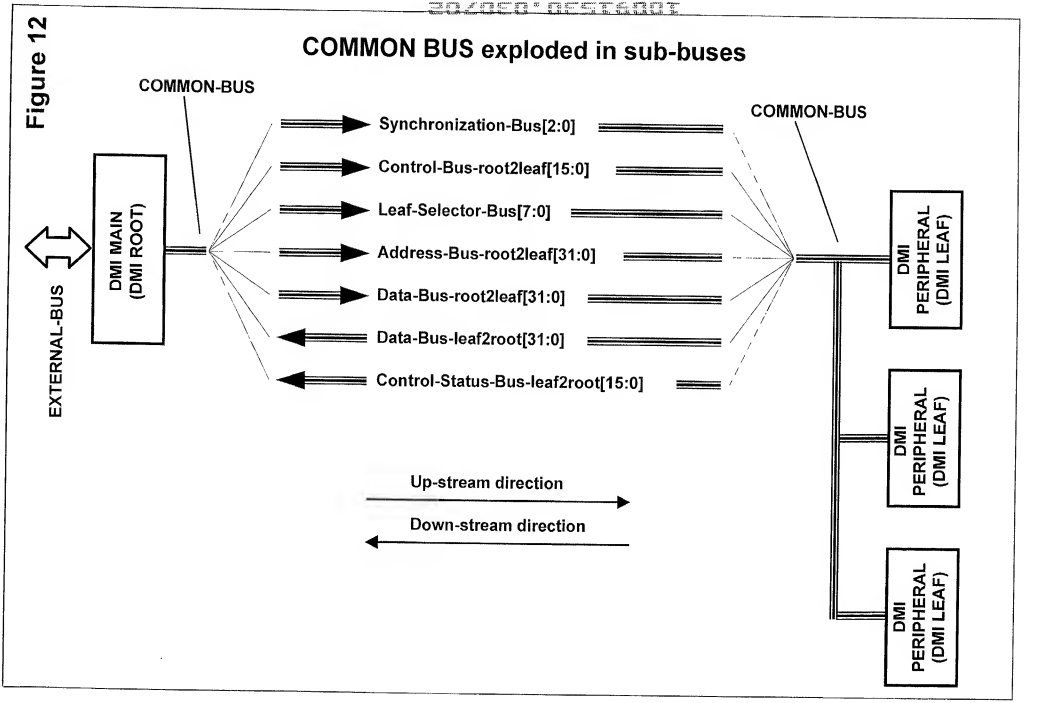


Figure 13

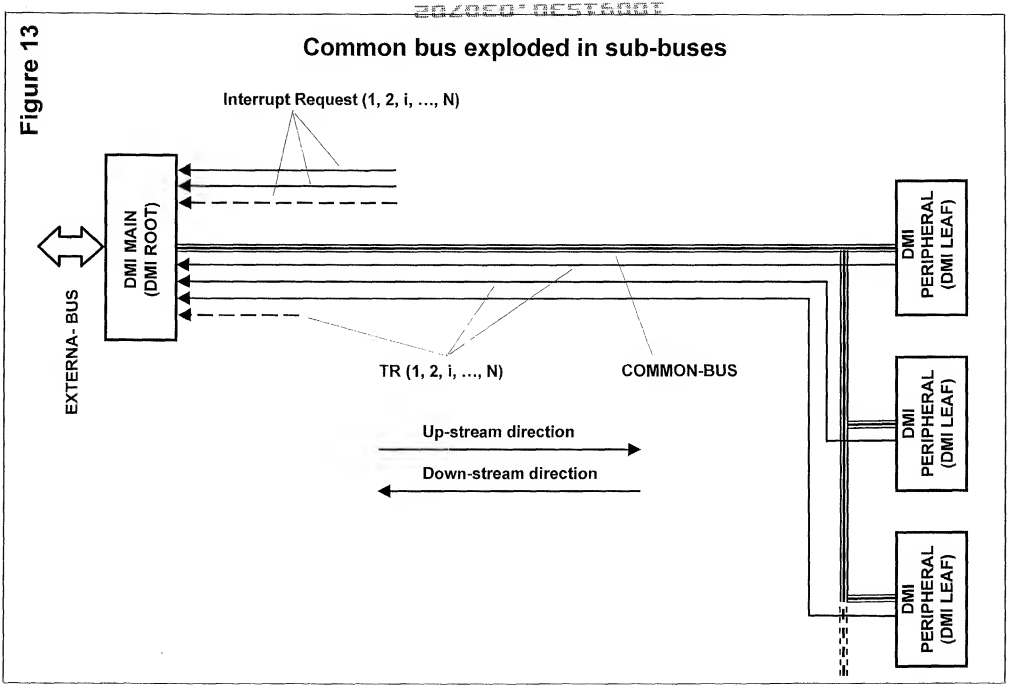

Common bus exploded in sub-buses
Interrupt Request (1, 2, i, ..., N)
DMI MAIN (DMI ROOT)
EXTERNA- BUS
TR (1, 2, i, ..., N)
COMMON-BUS
Up-stream direction
Down-stream direction
DMI PERIPHERAL (DMI LEAF)
DMI PERIPHERAL (DMI LEAF)
DMI PERIPHERAL (DMI LEAF)

Figure 14

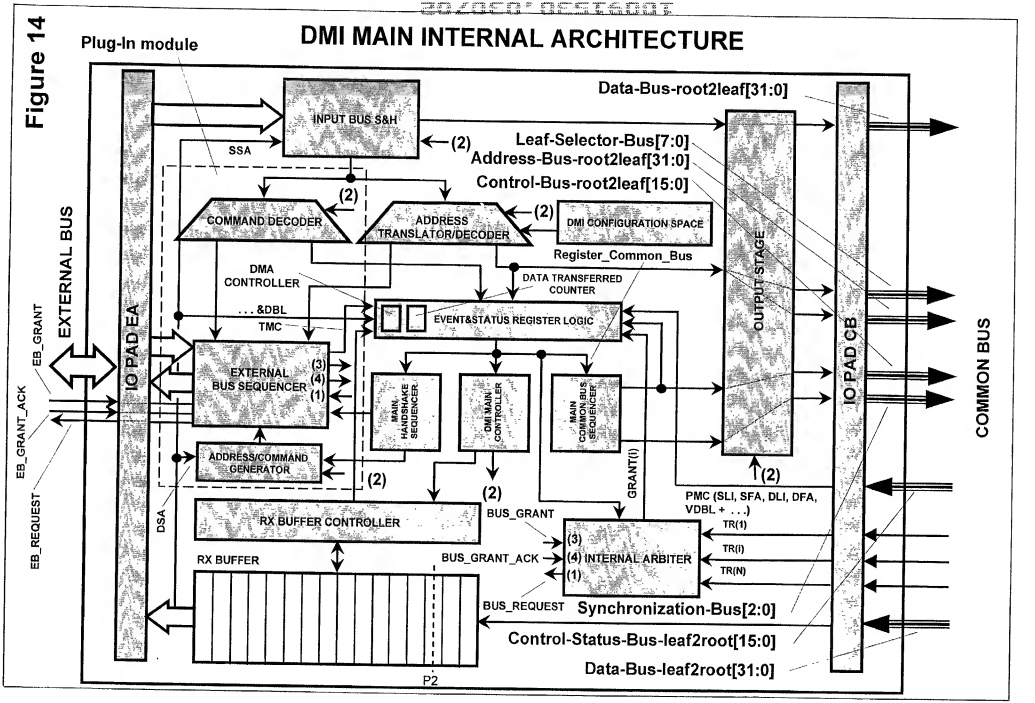

DMI MAIN INTERNAL ARCHITECTURE
Plug-In module
EXTERNAL BUS
IO PAD EA
EB_GRANT
EB_GRANT_ACK
EB_REQUEST
INPUT BUS S&H
SSA
(2)
Data-Bus-root2leaf[31:0]
Leaf-Selector-Bus[7:0]
Address-Bus-root2leaf[31:0]
Control-Bus-root2leaf[15:0]
COMMAND DECODER
ADDRESS TRANSLATOR/DECODER
DMI CONFIGURATION SPACE
Register_Common_Bus
DMA CONTROLLER
DATA TRANSFERRED COUNTER
... &DBL
TMC
EVENT&STATUS REGISTER LOGIC
OUTPUT STAGE
EXTERNAL BUS SEQUENCER
(3)
(4)
(1)
MAIN HANDSHAKE SEQUENCER
DMI MAIN CONTROLLER
MAIN COMMON BUS SEQUENCER
IO PAD CB
COMMON BUS
ADDRESS/COMMAND GENERATOR
GRANT(i)
PMC (SLI, SFA, DLI, DFA, VDBL + ...)
DSA
RX BUFFER CONTROLLER
BUS_GRANT
TR(1)
TR(i)
TR(N)
RX BUFFER
BUS_GRANT_ACK
INTERNAL ARBITER
BUS_REQUEST
Synchronization-Bus[2:0]
Control-Status-Bus-leaf2root[15:0]
Data-Bus-leaf2root[31:0]
P2

Figure 15

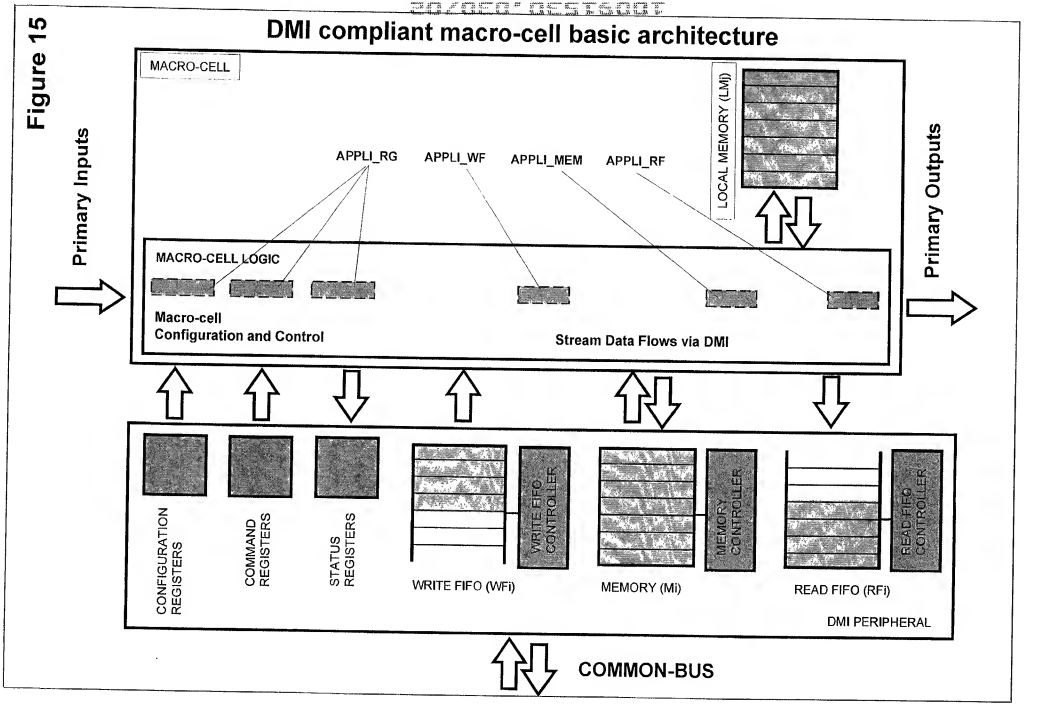

DMI compliant macro-cell basic architecture
MACRO-CELL
LOCAL MEMORY (LMI)
APPLI_RG
APPLI_WF
APPLI_MEM
APPLI_RF
Primary Inputs
Primary Outputs
MACRO-CELL LOGIC
Macro-cell
Configuration and Control
Stream Data Flows via DMI
CONFIGURATION REGISTERS
COMMAND REGISTERS
STATUS REGISTERS
WRITE FIFO CONTROLLER
MEMORY CONTROLLER
READ FIFO CONTROLLER
WRITE FIFO (WFi)
MEMORY (Mi)
READ FIFO (RFi)
DMI PERIPHERAL
COMMON-BUS

Figure 16

DMI PERIPHERAL Shadowed Layers

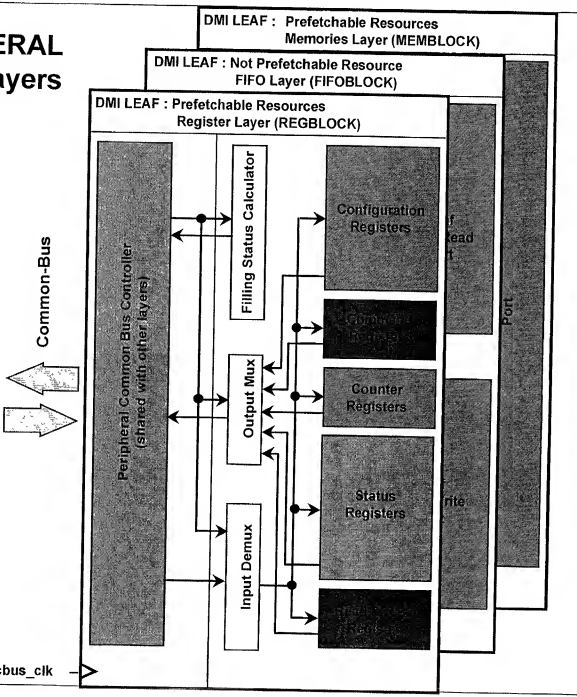


Figure 17

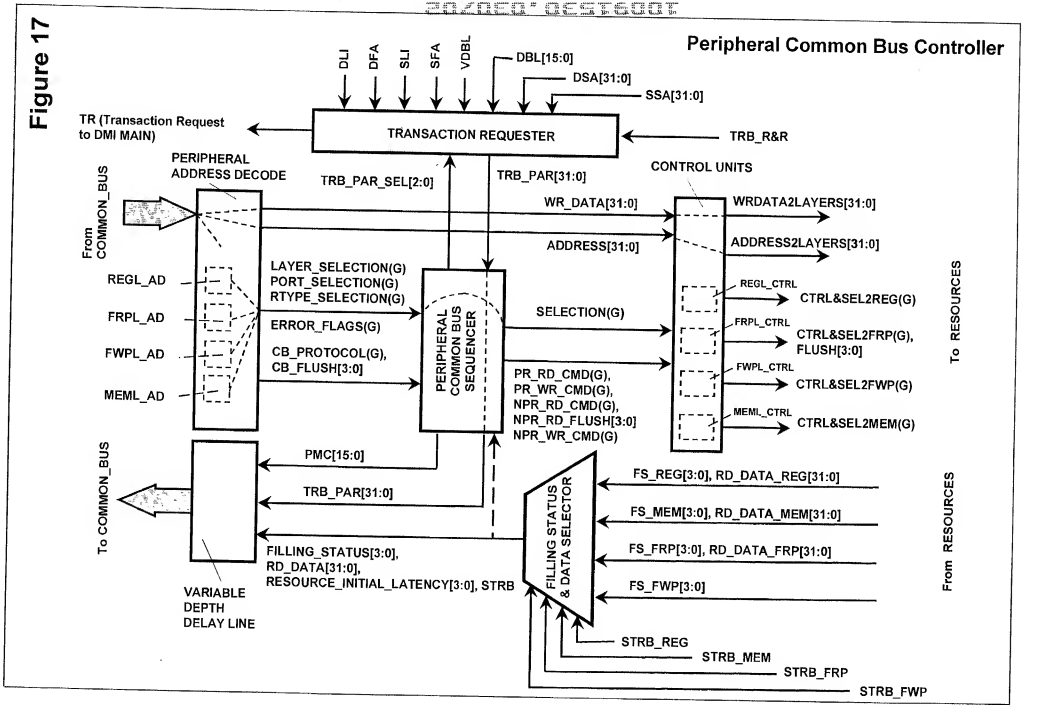

Peripheral Common Bus Controller
DLI
DFA
SLI
SFA
VDBL
DBL[15:0]
DSA[31:0]
SSA[31:0]
TR (Transaction Request to DMI MAIN)
TRANSACTION REQUESTER
TRB_R&R
PERIPHERAL ADDRESS DECODE
TRB_PAR_SEL[2:0]
TRB_PAR[31:0]
CONTROL UNITS
From COMMON_BUS
WR_DATA[31:0]
WRDATA2LAYERS[31:0]
ADDRESS[31:0]
ADDRESS2LAYERS[31:0]
REGL_AD
FRPL_AD
FWPL_AD
MEML_AD
LAYER_SELECTION(G)
PORT_SELECTION(G)
RTYPE_SELECTION(G)
ERROR_FLAGS(G)
CB_PROTOCOL(G),
CB_FLUSH[3:0]
PERIPHERAL COMMON BUS SEQUENCER
SELECTION(G)
PR_RD_CMD(G),
PR_WR_CMD(G),
NPR_RD_CMD(G),
NPR_RD_FLUSH[3:0]
NPR_WR_CMD(G)
REGL_CTRL
CTRL&SEL2REG(G)
FRPL_CTRL
CTRL&SEL2FRP(G),
FLUSH[3:0]
FWPL_CTRL
CTRL&SEL2FWP(G)
MEML_CTRL
CTRL&SEL2MEM(G)
To RESOURCES
To COMMON_BUS
PMC[15:0]
TRB_PAR[31:0]
FILLING STATUS & DATA SELECTOR
FS_REG[3:0], RD_DATA_REG[31:0]
FS_MEM[3:0], RD_DATA_MEM[31:0]
FS_FRP[3:0], RD_DATA_FRP[31:0]
FS_FWP[3:0]
From RESOURCES
FILLING_STATUS[3:0],
RD_DATA[31:0],
RESOURCE_INITIAL_LATENCY[3:0], STRB
VARIABLE DEPTH DELAY LINE
STRB_REG
STRB_MEM
STRB_FRP
STRB_FWP

Figure 18

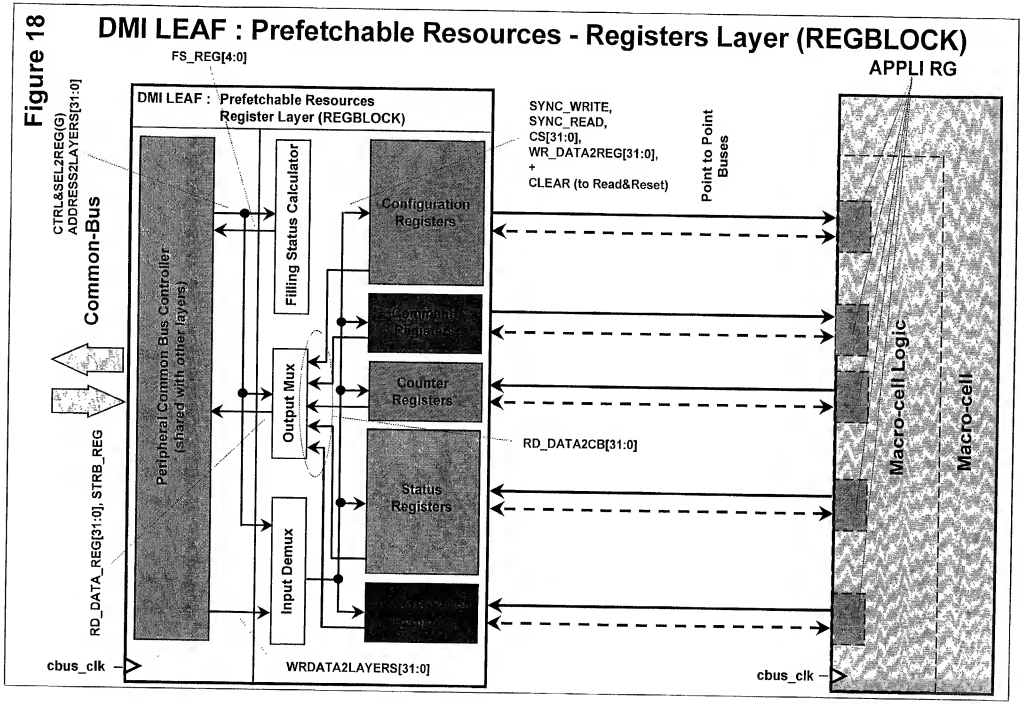

DMI LEAF : Prefetchable Resources - Registers Layer (REGBLOCK)
FS_REG[4:0]
APPLI RG
DMI LEAF : Prefetchable Resources Register Layer (REGBLOCK)
CTRL&SEL2REG(G)
ADDRESS2LAYERS[31:0]
Common-Bus
SYNC_WRITE,
SYNC_READ,
CS[31:0],
WR_DATA2REG[31:0],
+
CLEAR (to Read&Reset)
Point to Point Buses
Peripheral Common Bus Controller (shared with other layers)
Filling Status Calculator
Configuration Registers
Command Registers
Output Mux
Counter Registers
RD_DATA2CB[31:0]
Status Registers
Input Demux
RD_DATA_REG[31:0], STRB_REG
cbus_clk
WRDATA2LAYERS[31:0]
Macro-cell Logic
Macro-cell
cbus_clk

Figure 19

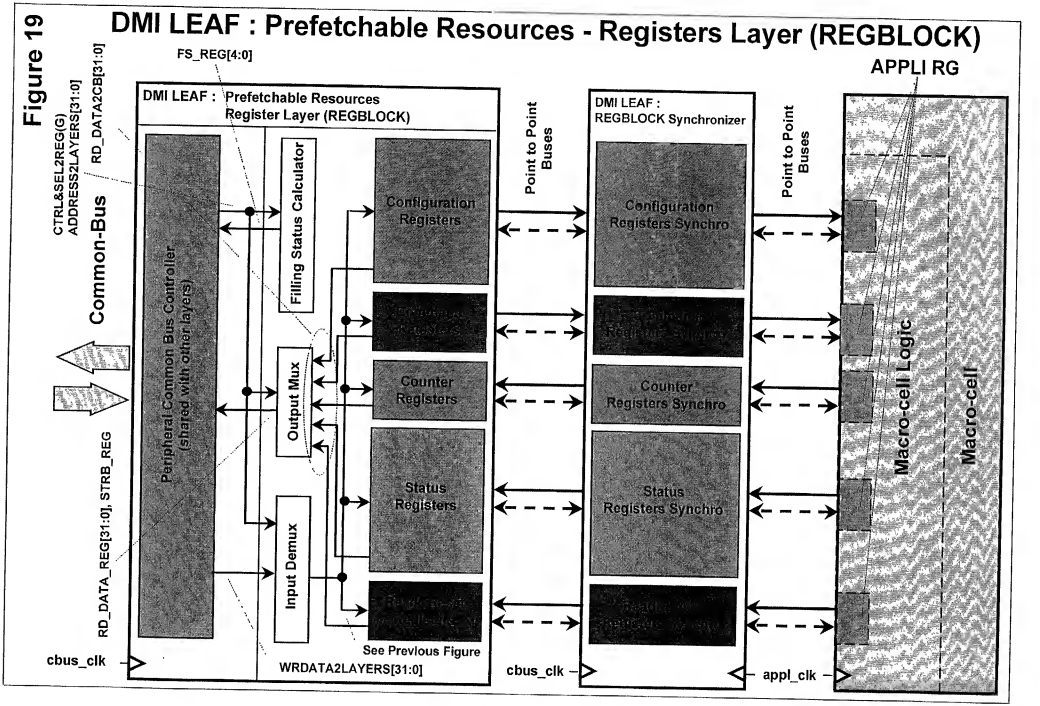
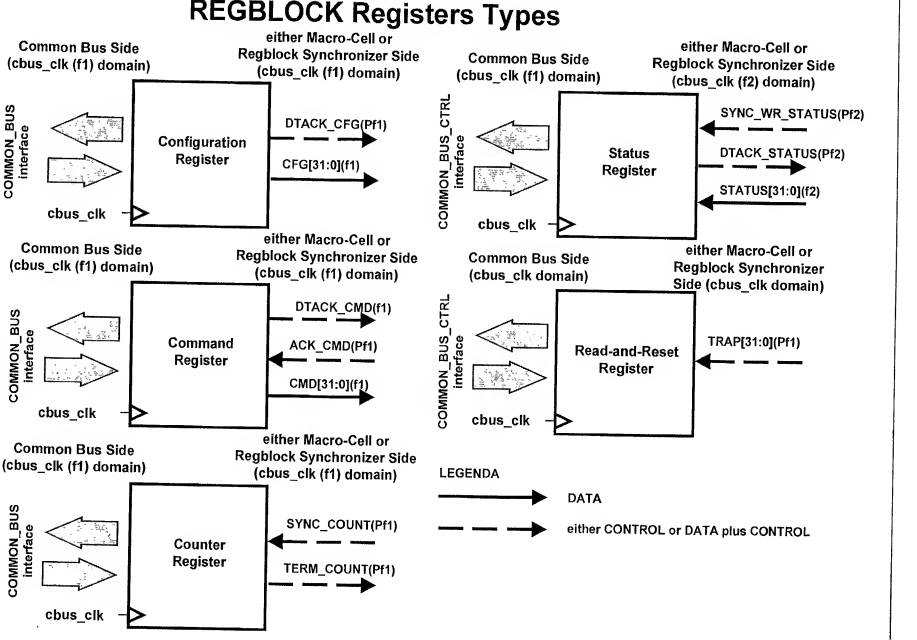


Figure 20



NOTE: referred to Counter Register:
DATA_FROM_CB[31:0] is THRESHOLD[31:0]
DATA2CB[31:0] is COUNTER[31:0]

Figure 21

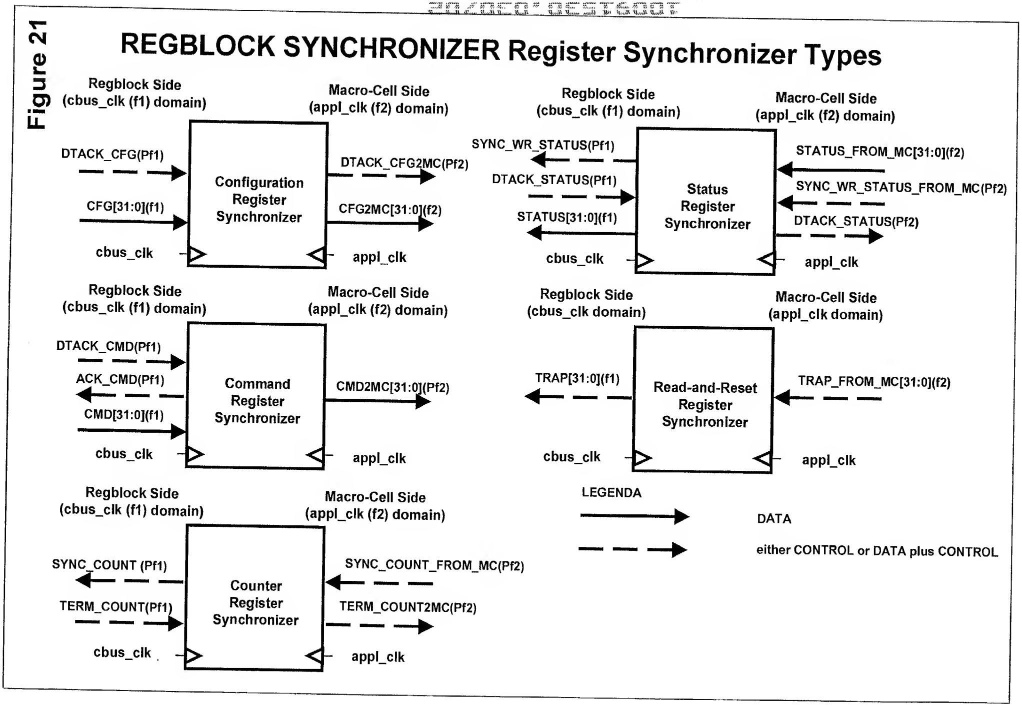

REGBLOCK SYNCHRONIZER Register Synchronizer Types
Regblock Side
(cbus_clk (f1) domain)
Macro-Cell Side
(appl_clk (f2) domain)
DTACK_CFG(Pf1)
CFG[31:0](f1)
cbus_clk
Configuration
Register
Synchronizer
DTACK_CFG2MC(Pf2)
CFG2MC[31:0](f2)
appl_clk
Regblock Side
(cbus_clk (f1) domain)
Macro-Cell Side
(appl_clk (f2) domain)
SYNC_WR_STATUS(Pf1)
DTACK_STATUS(Pf1)
STATUS[31:0](f1)
cbus_clk
Status
Register
Synchronizer
STATUS_FROM_MC[31:0](f2)
SYNC_WR_STATUS_FROM_MC(Pf2)
DTACK_STATUS(Pf2)
appl_clk
Regblock Side
(cbus_clk (f1) domain)
Macro-Cell Side
(appl_clk (f2) domain)
DTACK_CMD(Pf1)
ACK_CMD(Pf1)
CMD[31:0](f1)
cbus_clk
Command
Register
Synchronizer
CMD2MC[31:0](Pf2)
appl_clk
Regblock Side
(cbus_clk domain)
Macro-Cell Side
(appl_clk domain)
TRAP[31:0](f1)
cbus_clk
Read-and-Reset
Register
Synchronizer
TRAP_FROM_MC[31:0](f2)
appl_clk
Regblock Side
(cbus_clk (f1) domain)
Macro-Cell Side
(appl_clk (f2) domain)
SYNC_COUNT (Pf1)
TERM_COUNT(Pf1)
cbus_clk
Counter
Register
Synchronizer
SYNC_COUNT_FROM_MC(Pf2)
TERM_COUNT2MC(Pf2)
appl_clk
LEGENDA
DATA
either CONTROL or DATA plus CONTROL

Figure 22

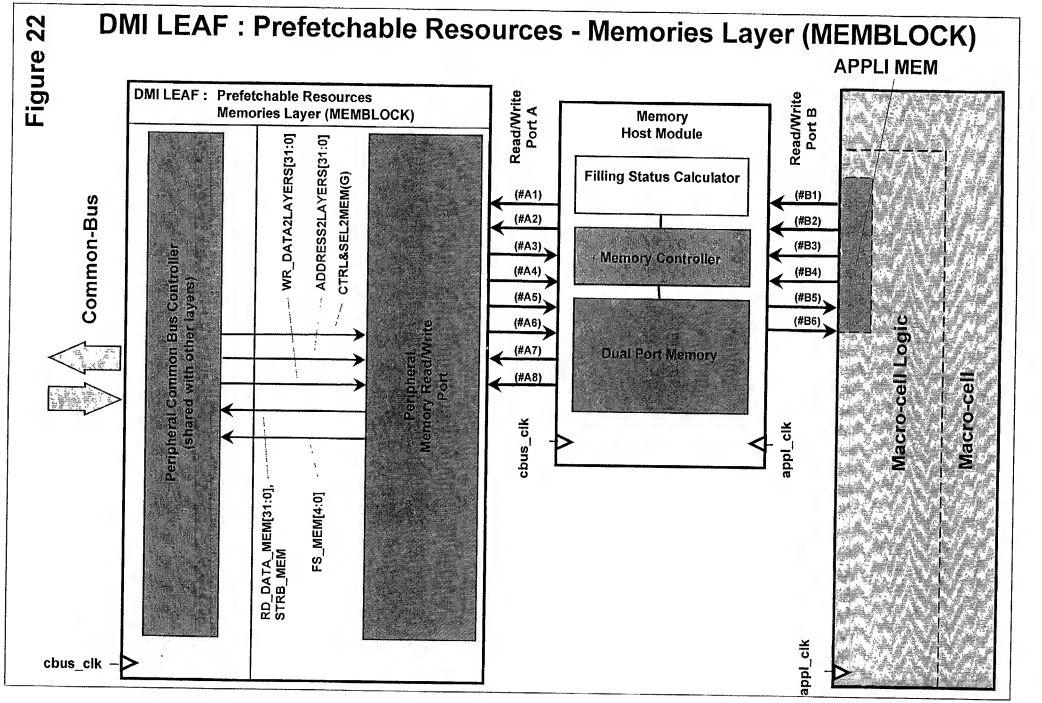


Figure 23

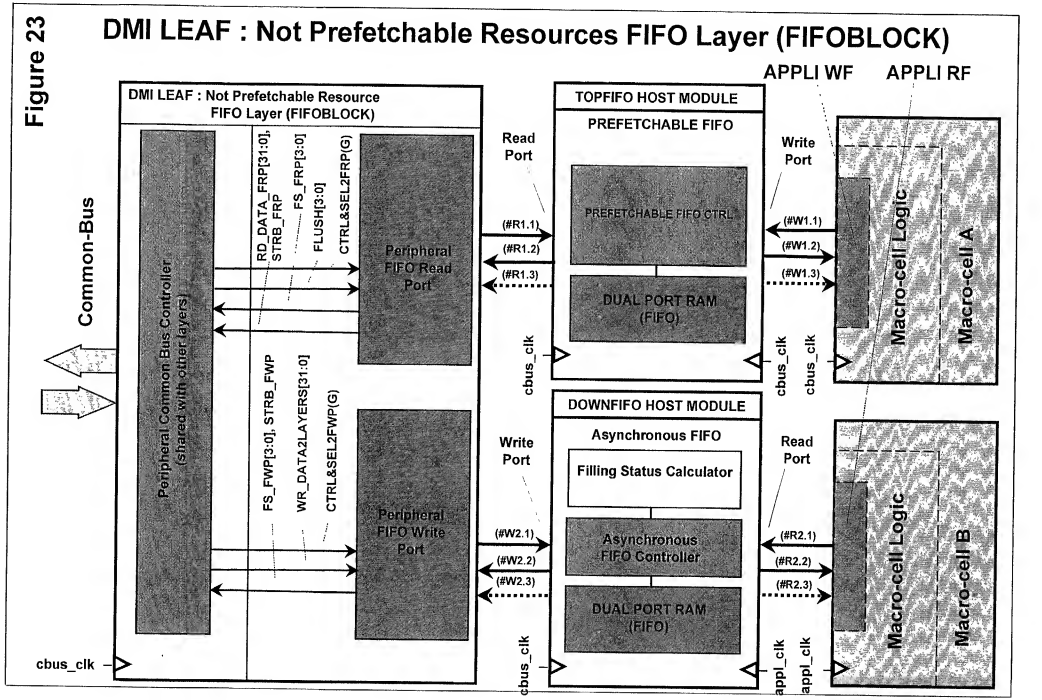


Figure 24

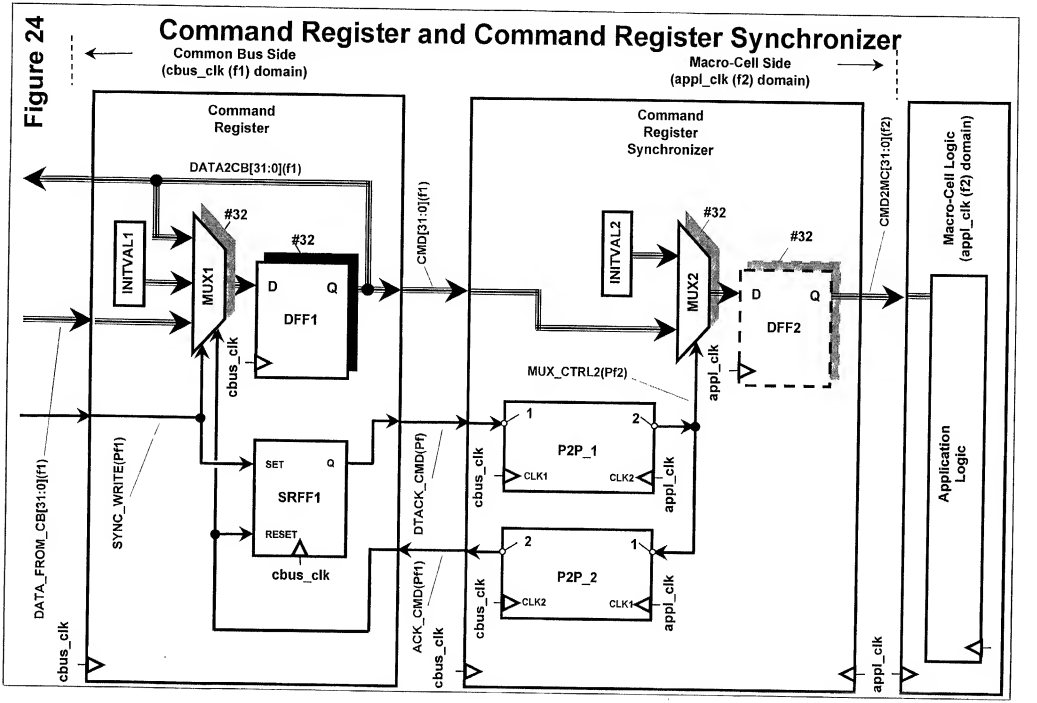


Figure 25

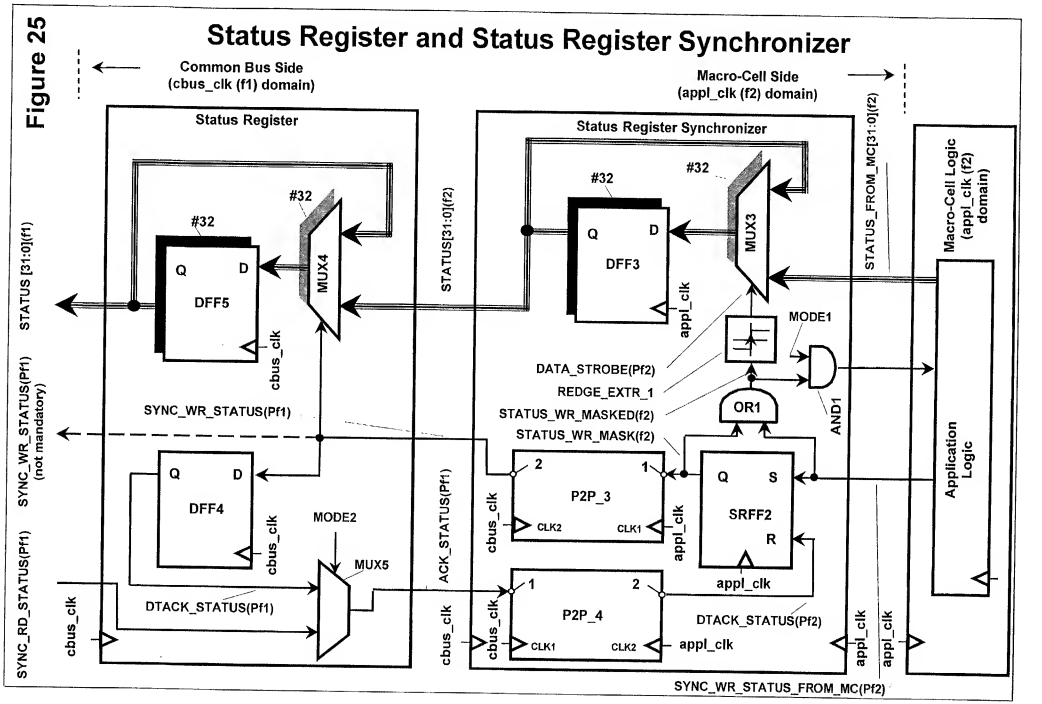


Figure 26

Pulse to Pulse Synchronization Unit

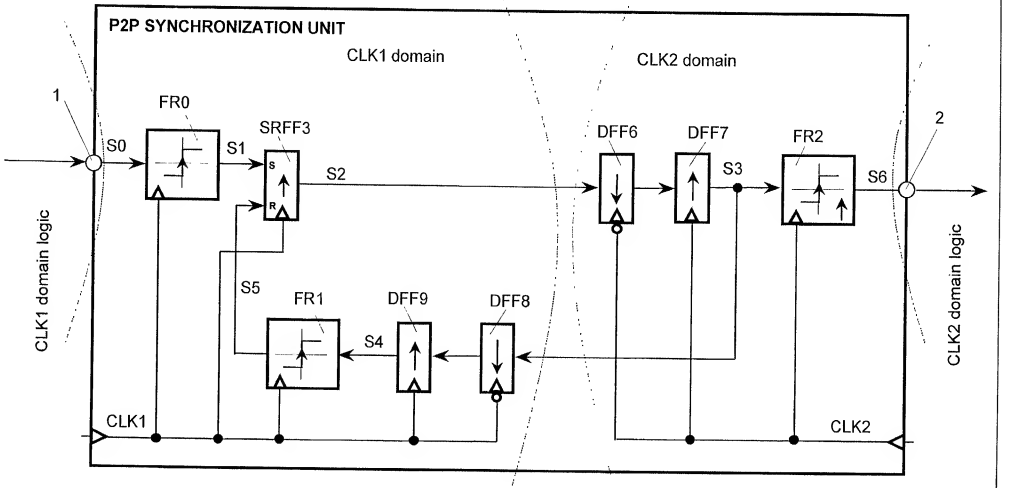


Figure 27

Timing Diagram of Pulse to Pulse Synchronization Unit

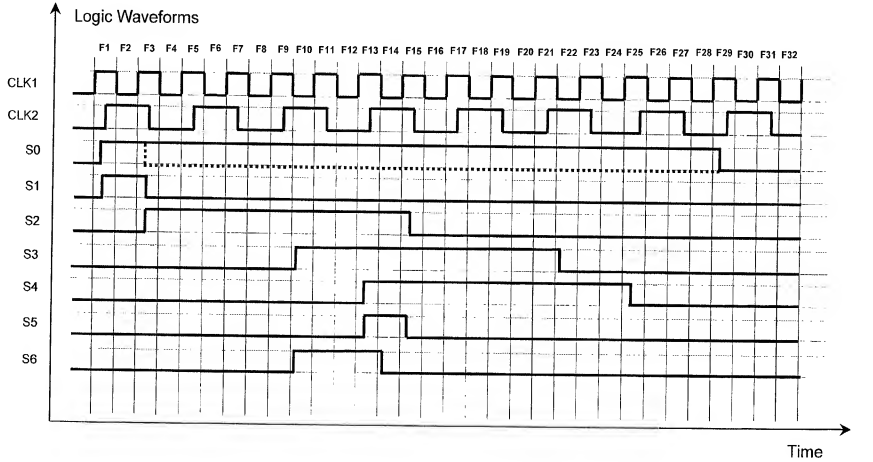


Figure 28

Advantages of Distributed Synchronization

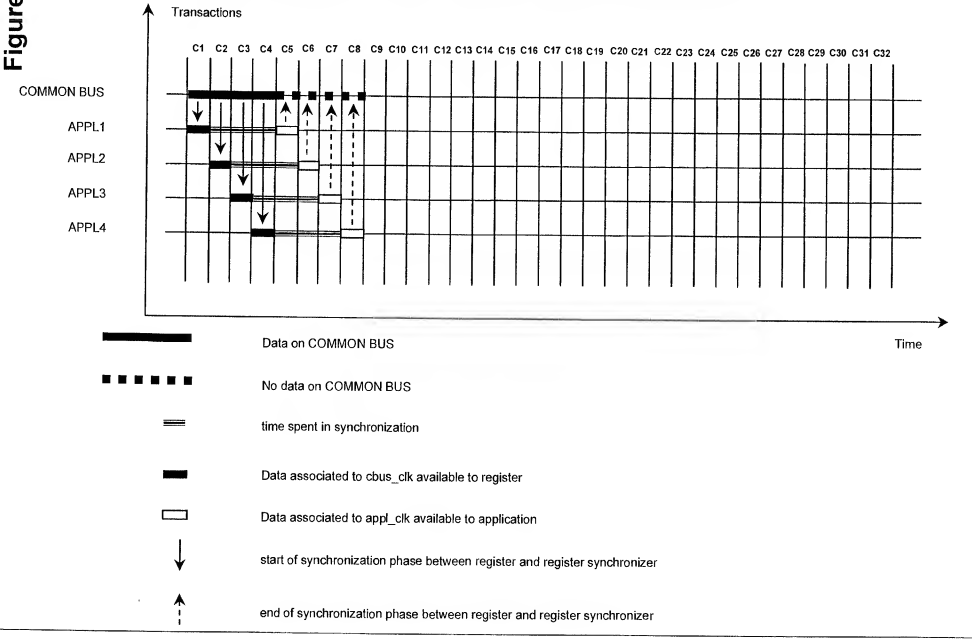


Figure29

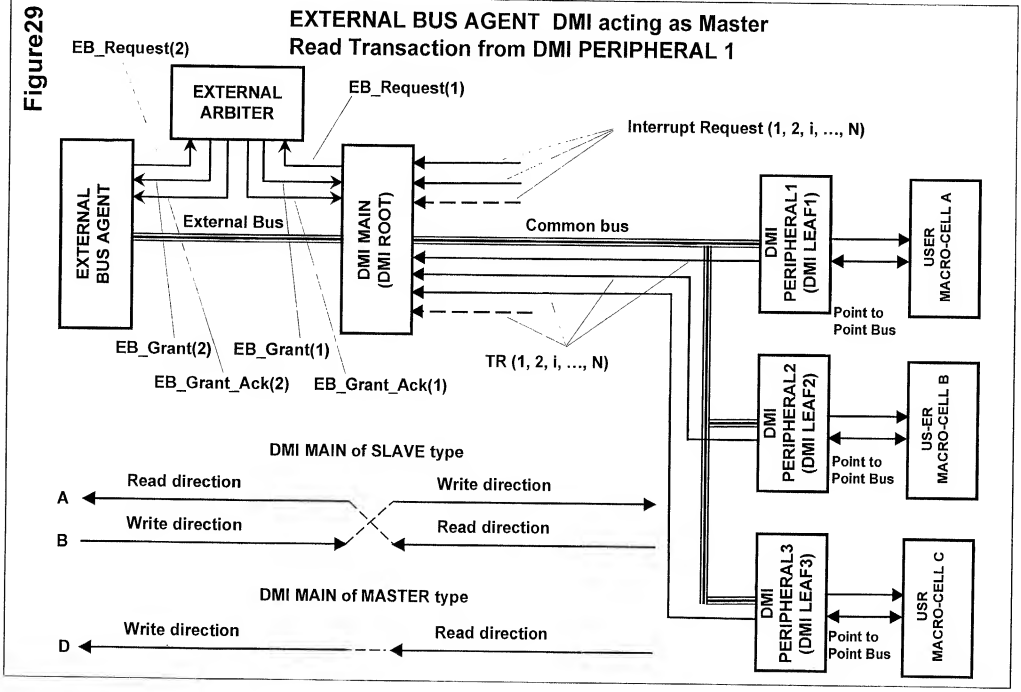


Figure 30

Prior Art

Asynchronous two phase handshake protocol: read

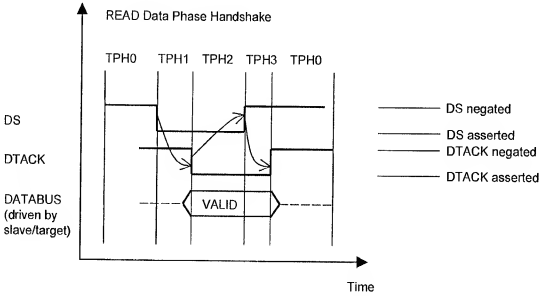


Figure 31

Prior Art

Asynchronous two phase handshake protocol: write

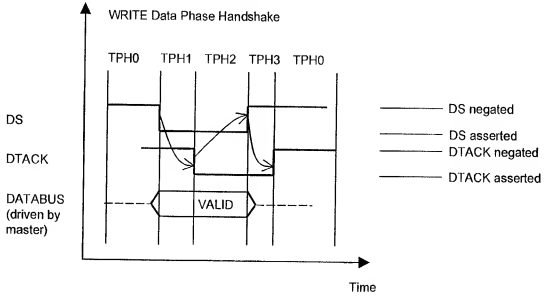
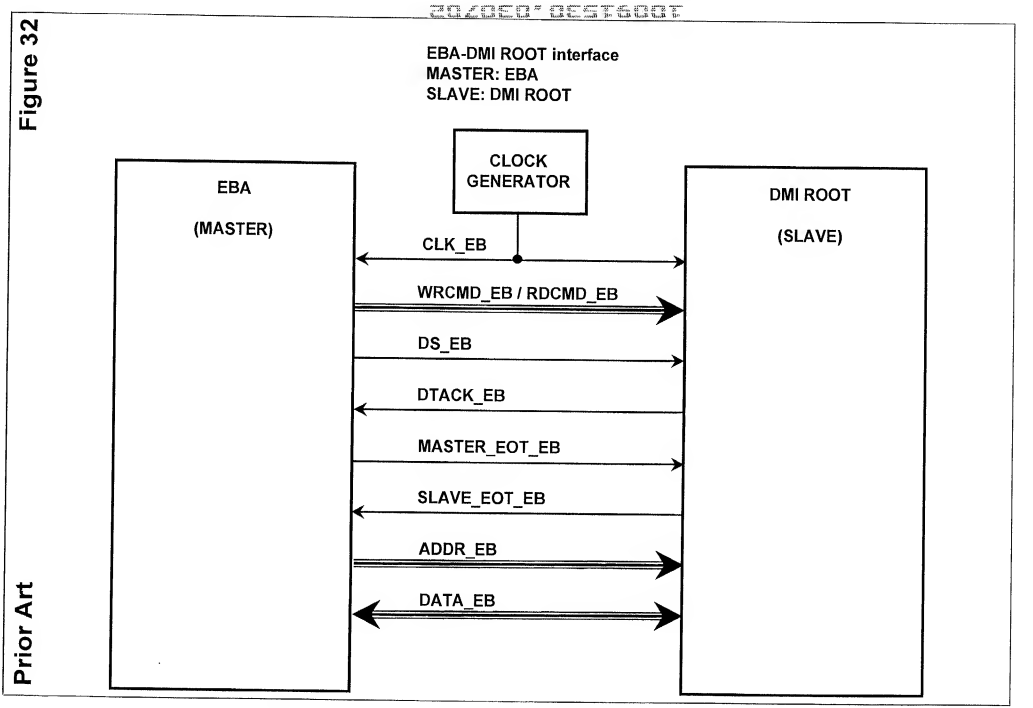


Figure 32



Prior Art

Figure 33

Prior Art

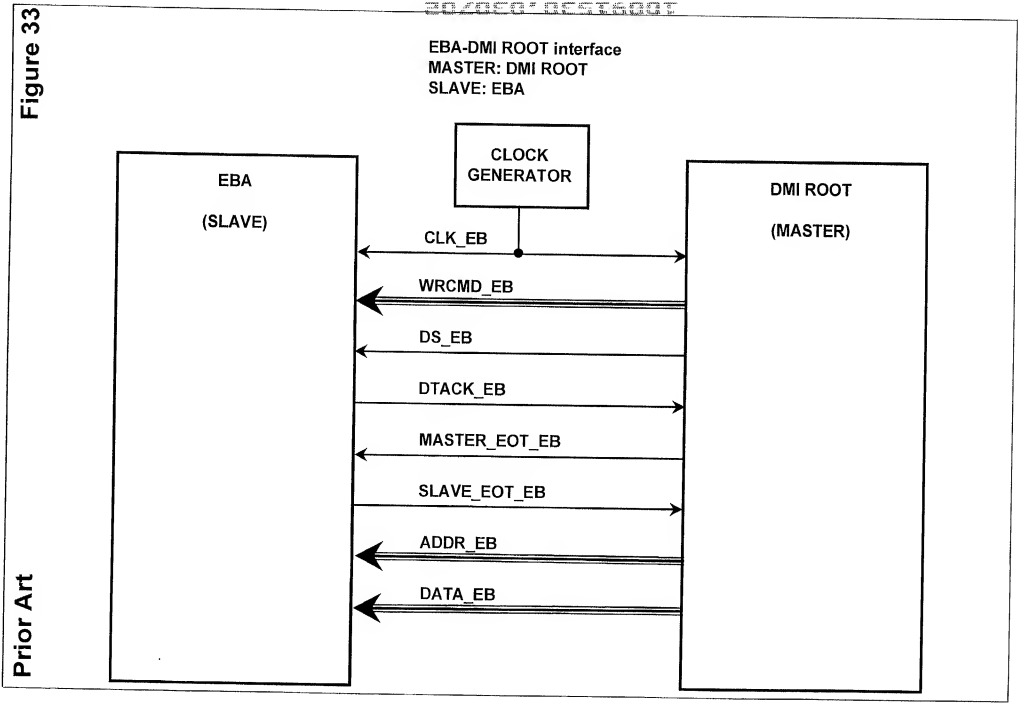


Figure 34

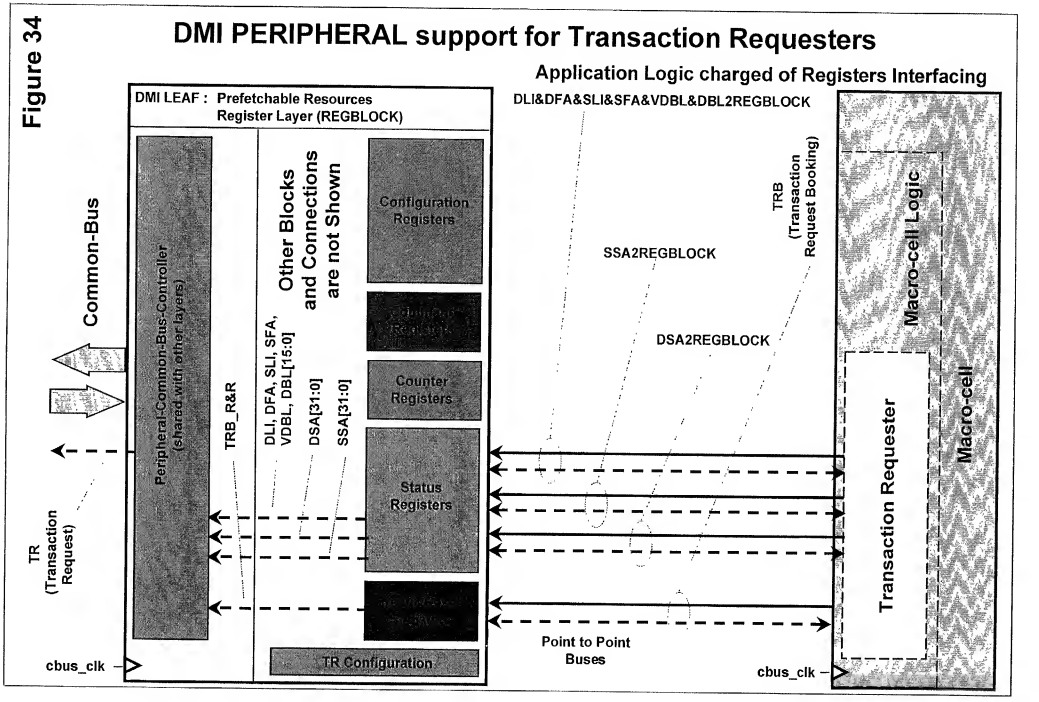


Figure 35

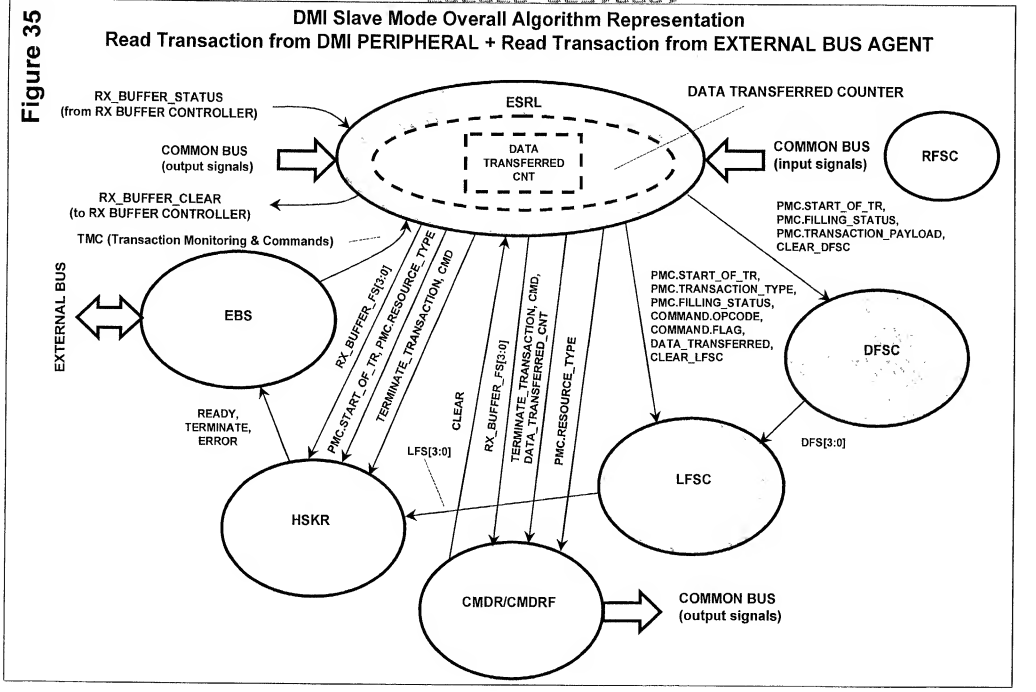

DMI Slave Mode Overall Algorithm Representation
Read Transaction from DMI PERIPHERAL + Read Transaction from EXTERNAL BUS AGENT
RX_BUFFER_STATUS (from RX BUFFER CONTROLLER)
COMMON BUS (output signals)
RX_BUFFER_CLEAR (to RX BUFFER CONTROLLER)
TMC (Transaction Monitoring & Commands)
EXTERNAL BUS
ESRL
DATA TRANSFERRED CNT
DATA TRANSFERRED COUNTER
COMMON BUS (input signals)
RFSC
PMC.START_OF_TR, PMC.FILLING_STATUS, PMC.TRANSACTION_PAYLOAD, CLEAR_DFSC
EBS
RX_BUFFER_FS[3:0]
PMC.START_OF_TR, PMC.RESOURCE_TYPE
TERMINATE_TRANSACTION, CMD
PMC.START_OF_TR, PMC.TRANSACTION_TYPE, PMC.FILLING_STATUS, COMMAND.OPCODE, COMMAND.FLAG, DATA_TRANSFERRED, CLEAR_LFSC
DFSC
READY, TERMINATE, ERROR
LFS[3:0]
CLEAR
RX_BUFFER_FS[3:0]
TERMINATE_TRANSACTION, CMD, DATA_TRANSFERRED_CNT
PMC.RESOURCE_TYPE
DFS[3:0]
LFSC
HSKR
CMDR/CMDRF
COMMON BUS (output signals)

Figure 36

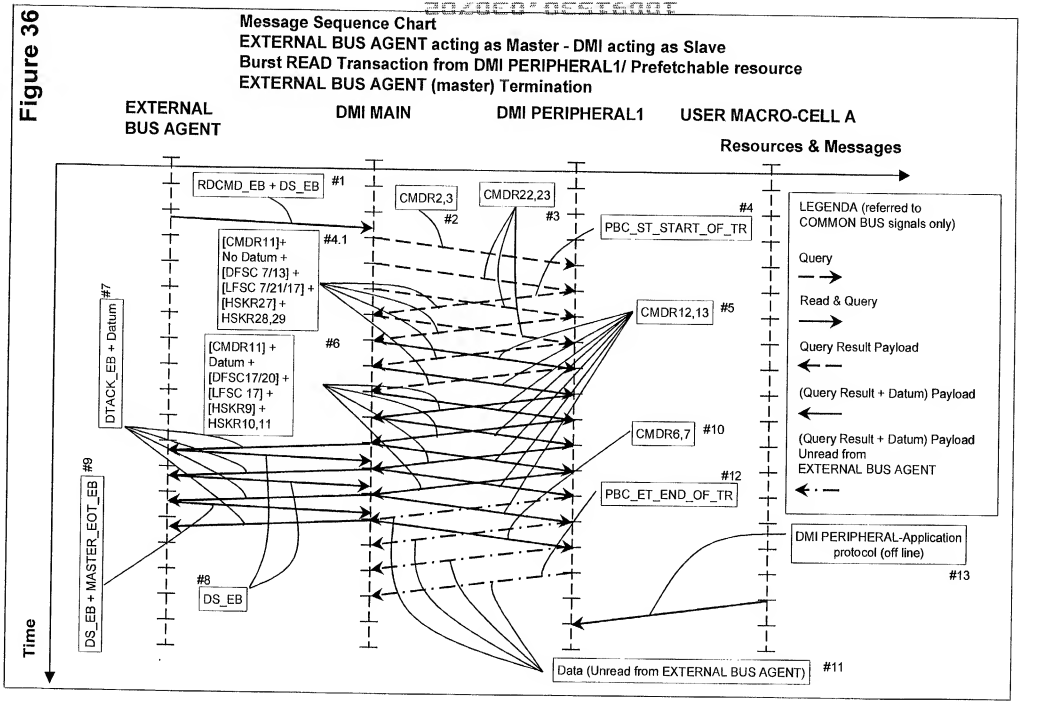
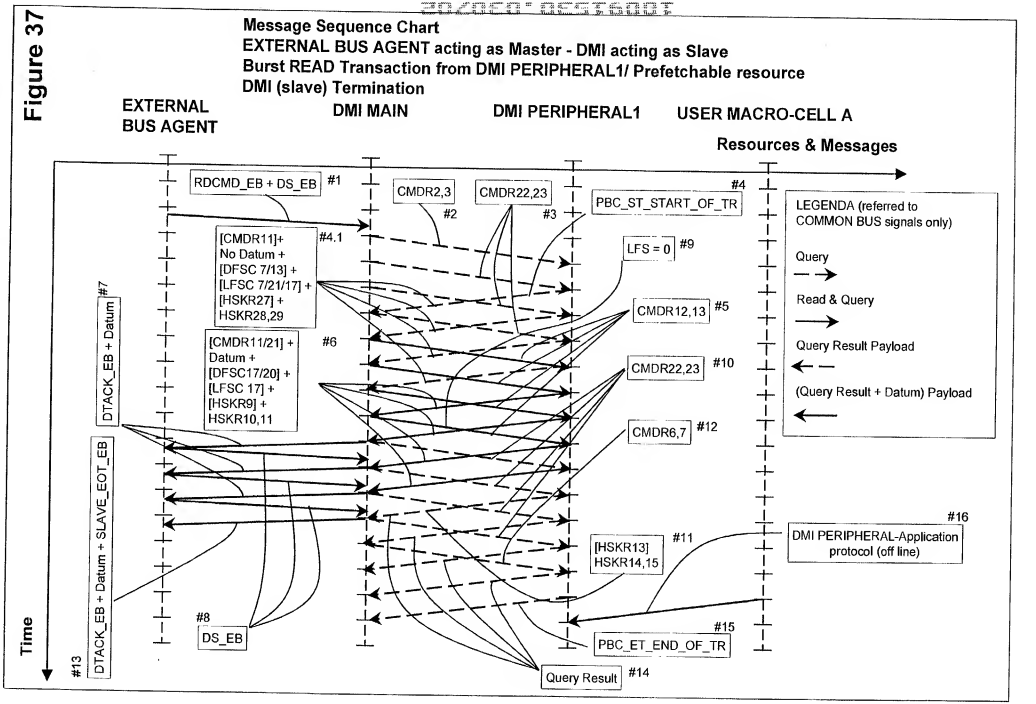

Message Sequence Chart
EXTERNAL BUS AGENT acting as Master - DMI acting as Slave
Burst READ Transaction from DMI PERIPHERAL1/ Prefetchable resource
EXTERNAL BUS AGENT (master) Termination
EXTERNAL BUS AGENT
DMI MAIN
DMI PERIPHERAL1
USER MACRO-CELL A
Resources & Messages
Time
RDCMD_EB + DS_EB
#1
CMDR2,3
#2
CMDR22,23
#3
#4
PBC_ST_START_OF_TR
[CMDR11]+ No Datum + [DFSC 7/13] + [LFSC 7/21/17] + [HSKR27] + HSKR28,29
#4.1
CMDR12,13
#5
[CMDR11] + Datum + [DFSC17/20] + [LFSC 17] + [HSKR9] + HSKR10,11
#6
DTACK_EB + Datum
#7
CMDR6,7
#10
#12
PBC_ET_END_OF_TR
DS_EB + MASTER_EOT_EB
#9
DMI PERIPHERAL-Application protocol (off line)
#13
#8
DS_EB
Data (Unread from EXTERNAL BUS AGENT)
#11
LEGENDA (referred to COMMON BUS signals only)
Query
Read & Query
Query Result Payload
(Query Result + Datum) Payload
(Query Result + Datum) Payload Unread from EXTERNAL BUS AGENT

Figure 37


Message Sequence Chart
EXTERNAL BUS AGENT acting as Master - DMI acting as Slave
Burst READ Transaction from DMI PERIPHERAL1/ Prefetchable resource
DMI (slave) Termination
EXTERNAL BUS AGENT
DMI MAIN
DMI PERIPHERAL1
USER MACRO-CELL A
Resources & Messages
Time
RDCMD_EB + DS_EB #1
CMDR2,3 #2
CMDR22,23 #3
PBC_ST_START_OF_TR #4
[CMDR11]+ No Datum + [DFSC 7/13] + [LFSC 7/21/17] + [HSKR27] + HSKR28,29 #4.1
LFS = 0 #9
CMDR12,13 #5
[CMDR11/21] + Datum + [DFSC17/20] + [LFSC 17] + [HSKR9] + HSKR10,11 #6
CMDR22,23 #10
DTACK_EB + Datum #7
CMDR6,7 #12
DTACK_EB + Datum + SLAVE_EOT_EB #13
[HSKR13] HSKR14,15 #11
DMI PERIPHERAL-Application protocol (off line) #16
DS_EB #8
PBC_ET_END_OF_TR #15
Query Result #14
LEGENDA (referred to COMMON BUS signals only)
Query
Read & Query
Query Result Payload
(Query Result + Datum) Payload

DMI Slave Mode Overall Algorithm Representation
Write Transaction from EXTERNAL BUS AGENT + Write Transaction to DMI PERIPHERAL

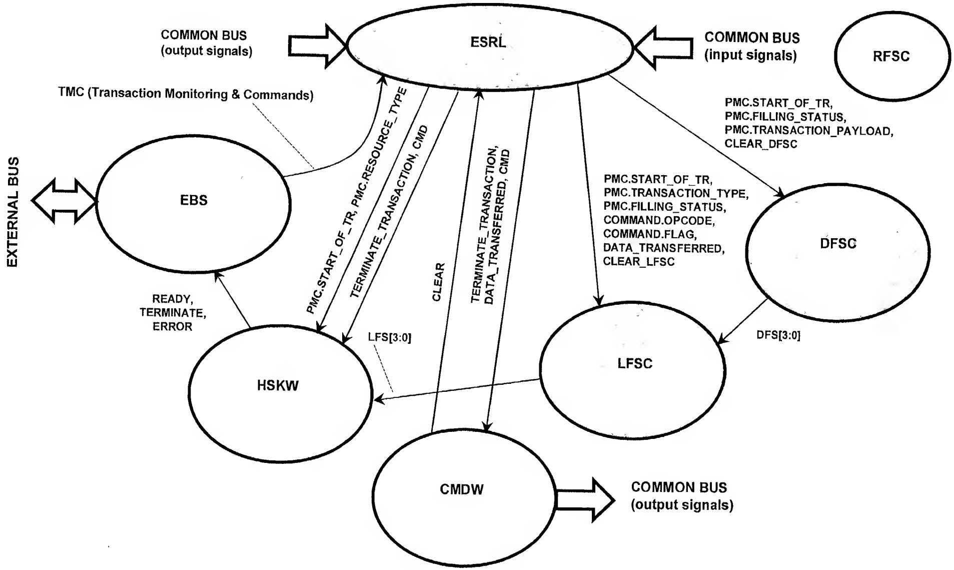
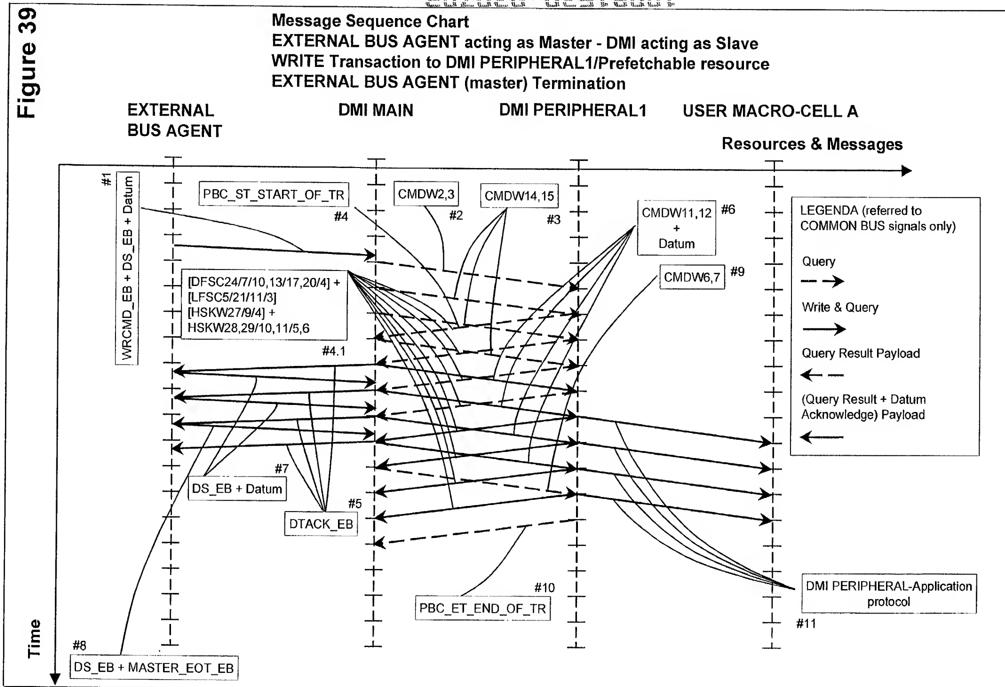


Figure 38

Figure 39



Message Sequence Chart
EXTERNAL BUS AGENT acting as Master - DMI acting as Slave
WRITE Transaction to DMI PERIPHERAL1/Prefetchable resource
EXTERNAL BUS AGENT (master) Termination
EXTERNAL BUS AGENT
DMI MAIN
DMI PERIPHERAL1
USER MACRO-CELL A
Resources & Messages
#1
WRCMD_EB + DS_EB + Datum
PBC_ST_START_OF_TR
#4
CMDW2,3
#2
CMDW14,15
#3
CMDW11,12 + Datum
#6
CMDW6,7
#9
[DFSC24/7/10,13/17,20/4] +
[LFSC5/21/11/3]
[HSKW27/9/4] +
HSKW28,29/10,11/5,6
#4.1
#7
DS_EB + Datum
#5
DTACK_EB
#10
PBC_ET_END_OF_TR
DMI PERIPHERAL-Application protocol
#11
#8
DS_EB + MASTER_EOT_EB
Time
LEGENDA (referred to COMMON BUS signals only)
Query
Write & Query
Query Result Payload
(Query Result + Datum Acknowledge) Payload

Figure 40

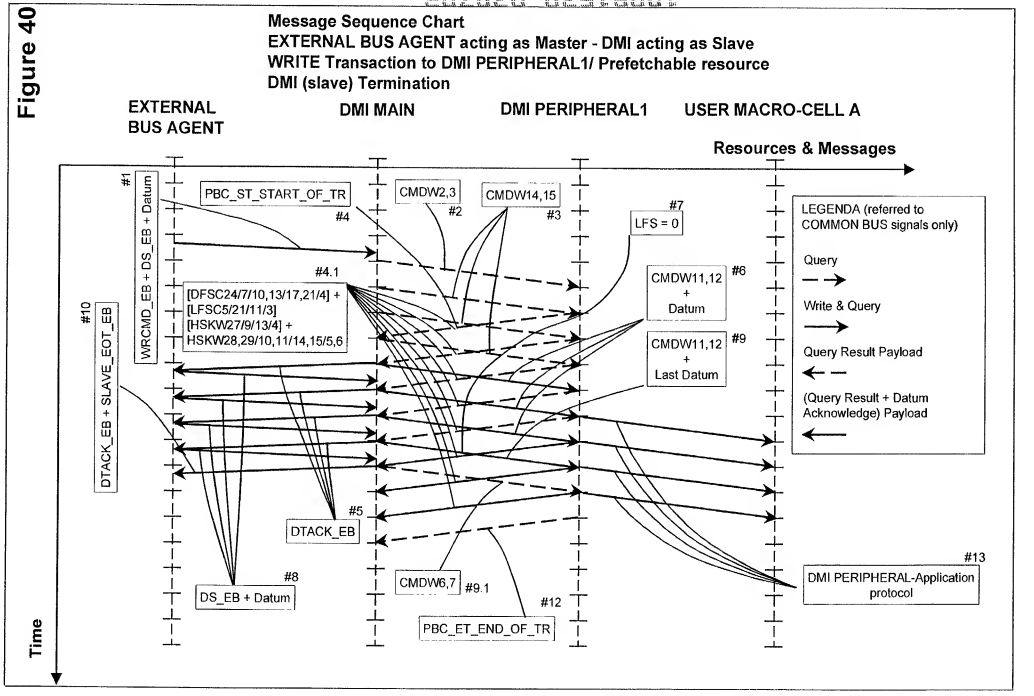


Figure 41

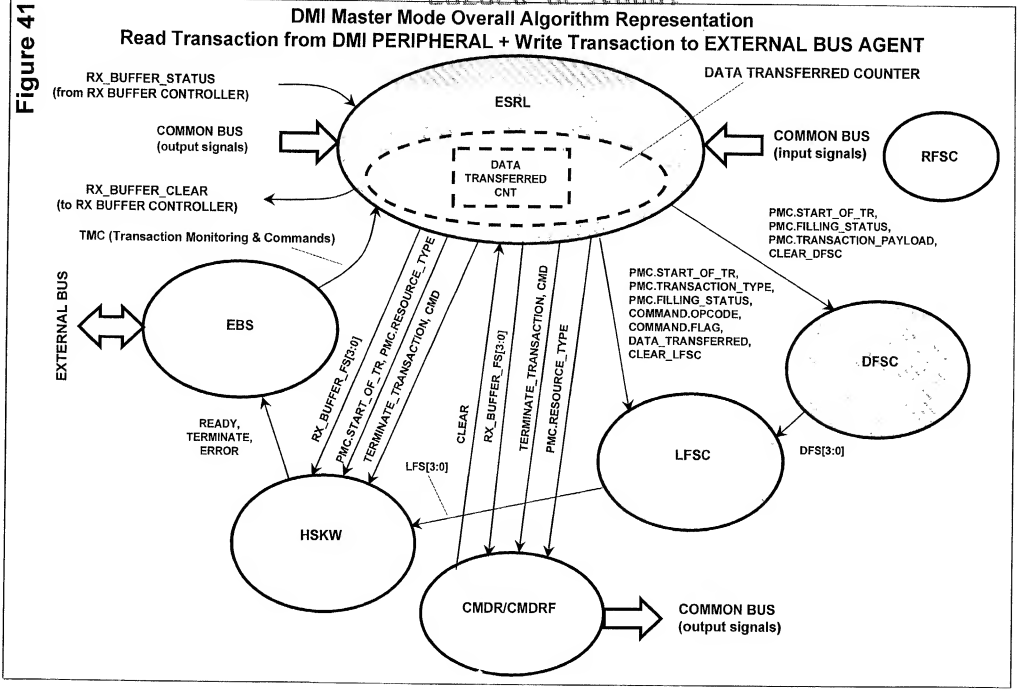

DMI Master Mode Overall Algorithm Representation
Read Transaction from DMI PERIPHERAL + Write Transaction to EXTERNAL BUS AGENT
RX_BUFFER_STATUS
(from RX BUFFER CONTROLLER)
COMMON BUS
(output signals)
RX_BUFFER_CLEAR
(to RX BUFFER CONTROLLER)
TMC (Transaction Monitoring & Commands)
EXTERNAL BUS
ESRL
DATA
TRANSFERRED
CNT
DATA TRANSFERRED COUNTER
COMMON BUS
(input signals)
RFSC
PMC.START_OF_TR,
PMC.FILLING_STATUS,
PMC.TRANSACTION_PAYLOAD,
CLEAR_DFSC
EBS
RX_BUFFER_FS[3:0]
PMC.START_OF_TR, PMC.RESOURCE_TYPE
TERMINATE_TRANSACTION, CMD
CLEAR
RX_BUFFER_FS[3:0]
TERMINATE_TRANSACTION, CMD
PMC.RESOURCE_TYPE
PMC.START_OF_TR,
PMC.TRANSACTION_TYPE,
PMC.FILLING_STATUS,
COMMAND.OPCODE,
COMMAND.FLAG,
DATA_TRANSFERRED,
CLEAR_LFSC
DFSC
READY,
TERMINATE,
ERROR
LFS[3:0]
LFSC
DFS[3:0]
HSKW
CMDR/CMDRF
COMMON BUS
(output signals)

Figure 42

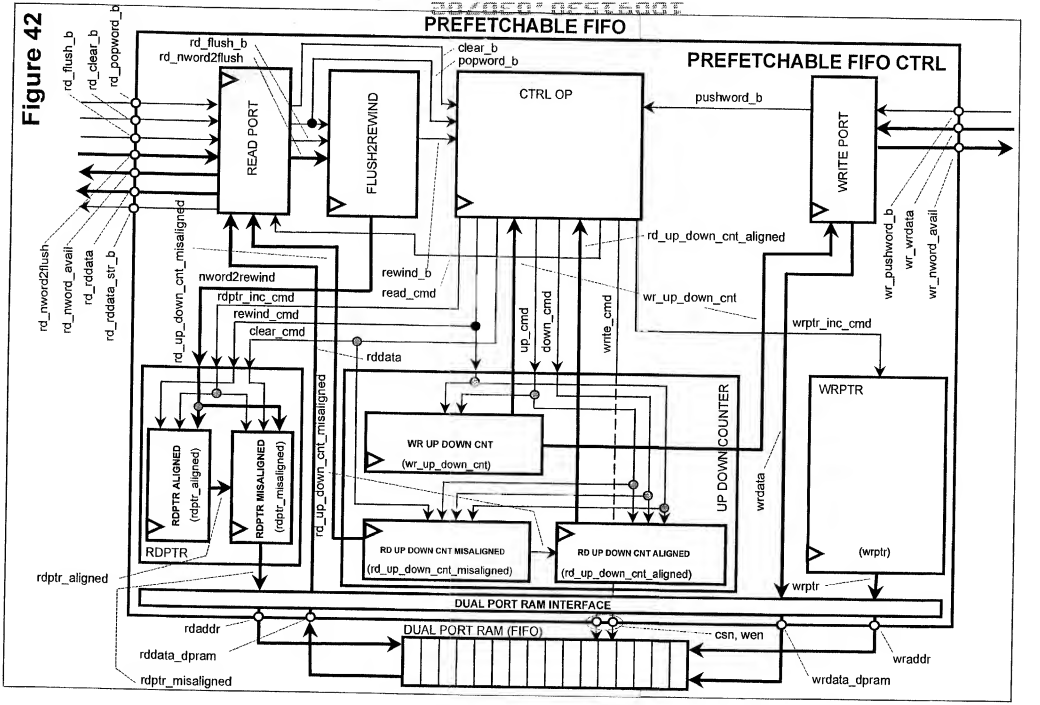

PREFETCHABLE FIFO
PREFETCHABLE FIFO CTRL
READ PORT
FLUSH2REWIND
CTRL OP
WRITE PORT
pushword_b
rd_flush_b
rd_clear_b
rd_popword_b
rd_nword2flush
rd_nword_avail
rd_rddata
rd_rddata_str_b
clear_b
popword_b
rd_up_down_cnt_misaligned
nword2rewind
rdptr_inc_cmd
rewind_cmd
clear_cmd
rewind_b
read_cmd
rddata
rd_up_down_cnt_aligned
wr_up_down_cnt
wrptr_inc_cmd
up_cmd
down_cmd
write_cmd
wr_pushword_b
wr_wrdata
wr_nword_avail
WR UP DOWN CNT
(wr_up_down_cnt)
UP DOWN COUNTER
wrdata
WRPTR
(wrptr)
RDPTR ALIGNED
(rdptr_aligned)
RDPTR MISALIGNED
(rdptr_misaligned)
RDPTR
RD UP DOWN CNT MISALIGNED
(rd_up_down_cnt_misaligned)
RD UP DOWN CNT ALIGNED
(rd_up_down_cnt_aligned)
rdptr_aligned
wrptr
DUAL PORT RAM INTERFACE
rdaddr
DUAL PORT RAM (FIFO)
csn, wen
wraddr
rddata_dpram
wrdata_dpram
rdptr_misaligned

Figure 43

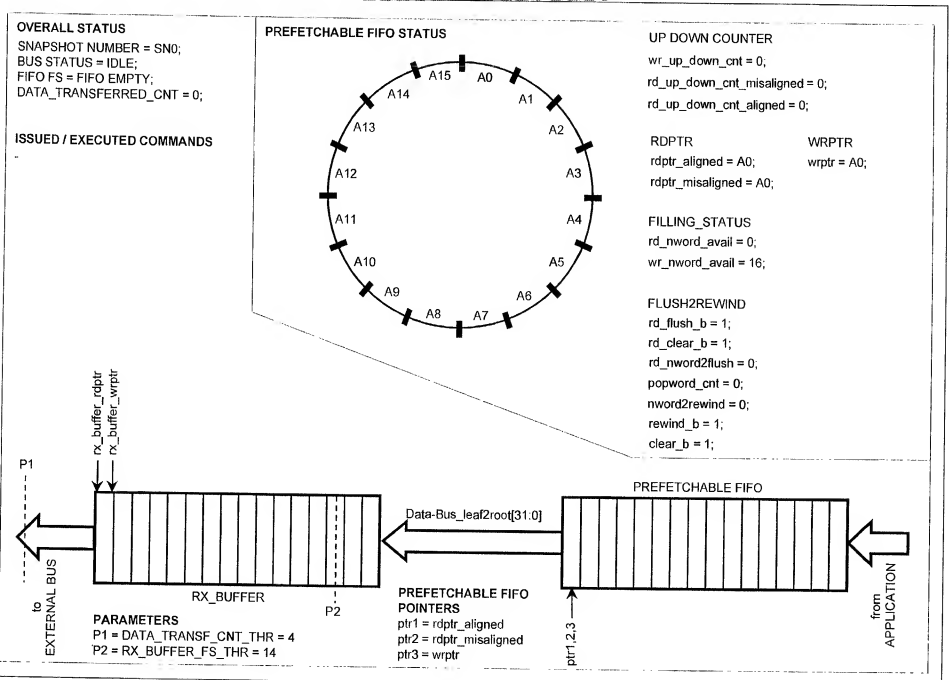

OVERALL STATUS
SNAPSHOT NUMBER = SN0;
BUS STATUS = IDLE;
FIFO FS = FIFO EMPTY;
DATA_TRANSFERRED_CNT = 0;
ISSUED / EXECUTED COMMANDS
PREFETCHABLE FIFO STATUS
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
UP DOWN COUNTER
wr_up_down_cnt = 0;
rd_up_down_cnt_misaligned = 0;
rd_up_down_cnt_aligned = 0;
RDPTR
rdptr_aligned = A0;
rdptr_misaligned = A0;
WRPTR
wrptr = A0;
FILLING_STATUS
rd_nword_avail = 0;
wr_nword_avail = 16;
FLUSH2REWIND
rd_flush_b = 1;
rd_clear_b = 1;
rd_nword2flush = 0;
popword_cnt = 0;
nword2rewind = 0;
rewind_b = 1;
clear_b = 1;
rx_buffer_rdptr
rx_buffer_wrptr
P1
PREFETCHABLE FIFO
Data-Bus_leaf2root[31:0]
to EXTERNAL BUS
RX_BUFFER
P2
from APPLICATION
PARAMETERS
P1 = DATA_TRANSF_CNT_THR = 4
P2 = RX_BUFFER_FS_THR = 14
PREFETCHABLE FIFO POINTERS
ptr1 = rdptr_aligned
ptr2 = rdptr_misaligned
ptr3 = wrptr
ptr1,2,3

Figure 44

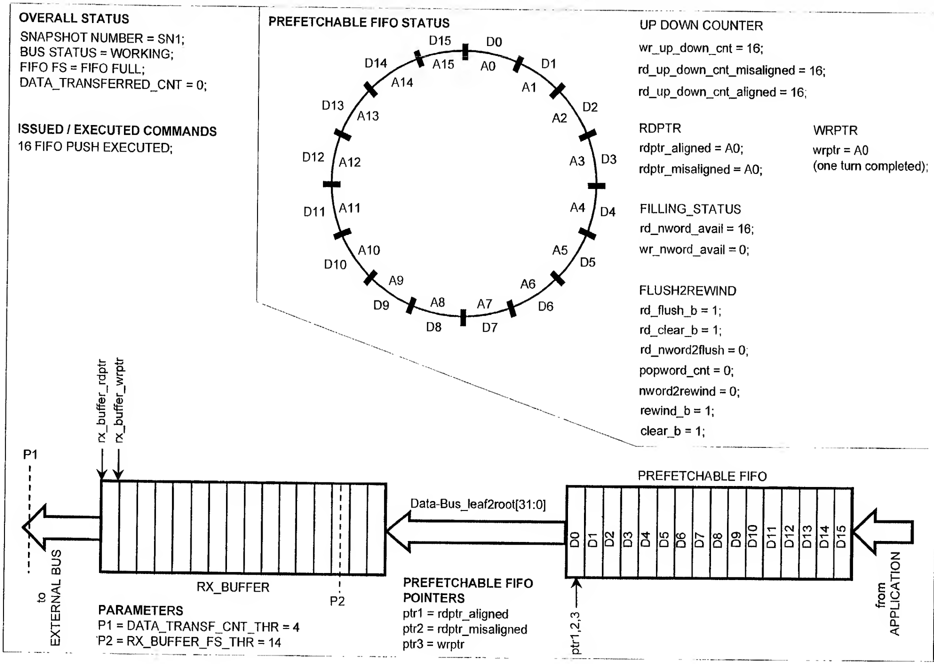

OVERALL STATUS
SNAPSHOT NUMBER = SN1;
BUS STATUS = WORKING;
FIFO FS = FIFO FULL;
DATA_TRANSFERRED_CNT = 0;
ISSUED / EXECUTED COMMANDS
16 FIFO PUSH EXECUTED;
PREFETCHABLE FIFO STATUS
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15
UP DOWN COUNTER
wr_up_down_cnt = 16;
rd_up_down_cnt_misaligned = 16;
rd_up_down_cnt_aligned = 16;
RDPTR
rdptr_aligned = A0;
rdptr_misaligned = A0;
WRPTR
wrptr = A0
(one turn completed);
FILLING_STATUS
rd_nword_avail = 16;
wr_nword_avail = 0;
FLUSH2REWIND
rd_flush_b = 1;
rd_clear_b = 1;
rd_nword2flush = 0;
popword_cnt = 0;
nword2rewind = 0;
rewind_b = 1;
clear_b = 1;
P1
rx_buffer_rdptr
rx_buffer_wrptr
to EXTERNAL BUS
RX_BUFFER
P2
Data-Bus_leaf2root[31:0]
PREFETCHABLE FIFO
ptr1,2,3
from APPLICATION
PARAMETERS
P1 = DATA_TRANSF_CNT_THR = 4
P2 = RX_BUFFER_FS_THR = 14
PREFETCHABLE FIFO POINTERS
ptr1 = rdptr_aligned
ptr2 = rdptr_misaligned
ptr3 = wrptr

Figure 45

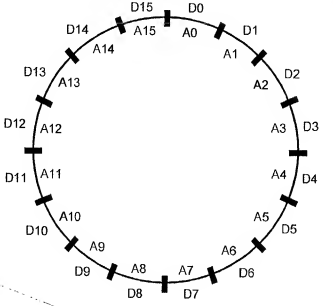
OVERALL STATUS

SNAPSHOT NUMBER = SN2;
BUS STATUS = WORKING;
FIFO FS = FIFO FULL;
DATA_TRANSFERRED_CNT = 0;

ISSUED / EXECUTED COMMANDS

7 POP EXECUTED;

PREFETCHABLE FIFO STATUS



UP DOWN COUNTER

wr_up_down_cnt = 16;
rd_up_down_cnt_misaligned = 9;
rd_up_down_cnt_aligned = 9;

RDPTR

rdptr_aligned = A7;
rdptr_misaligned = A7;

WRPTR

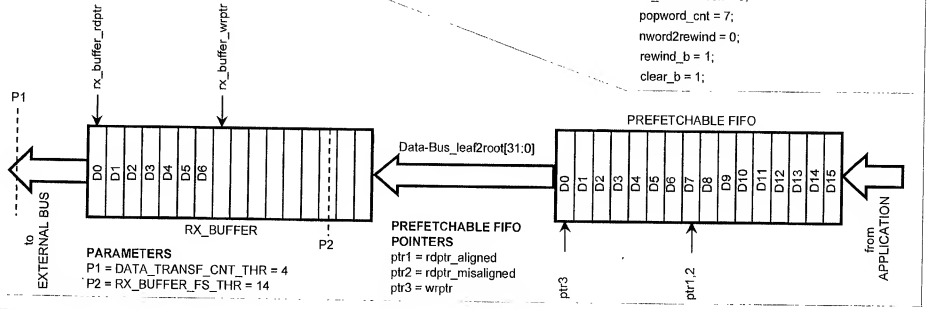
wrptr = A0;

FILLING_STATUS

rd_nword_avail = 9;
wr_nword_avail = 0;

FLUSH2REWIND

rd_flush_b = 1;
rd_clear_b = 1;
rd_nword2flush = 0;
popword_cnt = 7;
nword2rewind = 0;
rewind_b = 1;
clear_b = 1;



PARAMETERS

P1 = DATA_TRANSF_CNT_THR = 4
P2 = RX_BUFFER_FS_THR = 14

PREFETCHABLE FIFO POINTERS

ptr1 = rdptr_aligned
ptr2 = rdptr_misaligned
ptr3 = wrptr

Figure 46

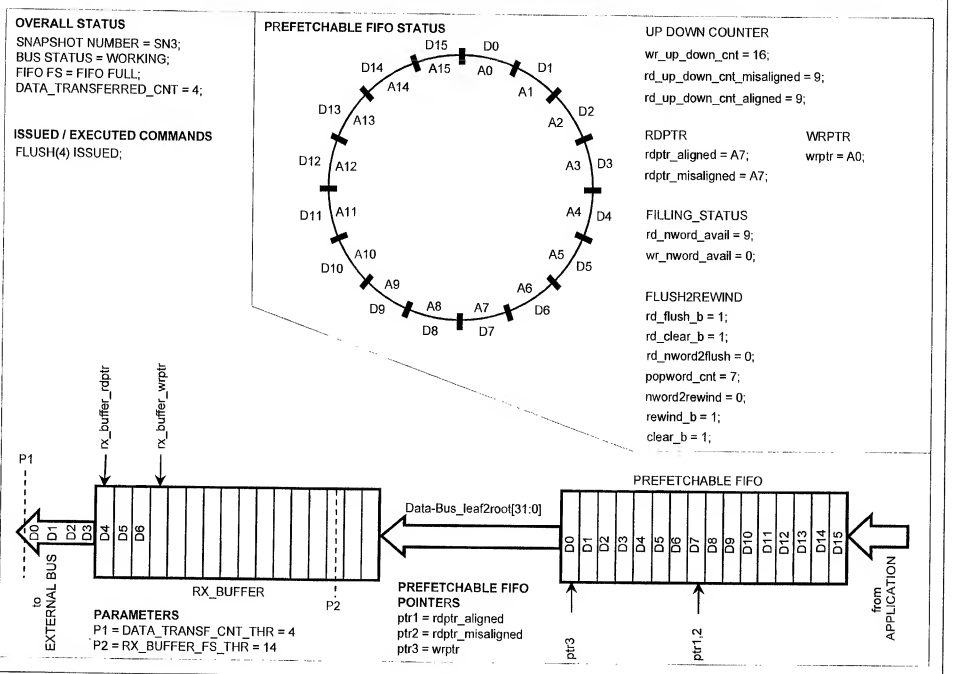

OVERALL STATUS
SNAPSHOT NUMBER = SN3;
BUS STATUS = WORKING;
FIFO FS = FIFO FULL;
DATA_TRANSFERRED_CNT = 4;
ISSUED / EXECUTED COMMANDS
FLUSH(4) ISSUED;
PREFETCHABLE FIFO STATUS
D15 D0 D14 A15 A0 D1 A14 A1 D13 D2 A13 A2 D12 A3 D3 A12 D11 A4 D4 A11 D10 A5 D5 A10 D9 A6 D6 A9 D8 A8 A7 D7
UP DOWN COUNTER
wr_up_down_cnt = 16;
rd_up_down_cnt_misaligned = 9;
rd_up_down_cnt_aligned = 9;
RDPTR
rdptr_aligned = A7;
rdptr_misaligned = A7;
WRPTR
wrptr = A0;
FILLING_STATUS
rd_nword_avail = 9;
wr_nword_avail = 0;
FLUSH2REWIND
rd_flush_b = 1;
rd_clear_b = 1;
rd_nword2flush = 0;
popword_cnt = 7;
nword2rewind = 0;
rewind_b = 1;
clear_b = 1;
P1
rx_buffer_rdptr
rx_buffer_wrptr
D0 D1 D2 D3 D4 D5 D6
RX_BUFFER
P2
to EXTERNAL BUS
PARAMETERS
P1 = DATA_TRANSF_CNT_THR = 4
P2 = RX_BUFFER_FS_THR = 14
Data-Bus_leaf2root[31:0]
PREFETCHABLE FIFO
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15
PREFETCHABLE FIFO POINTERS
ptr1 = rdptr_aligned
ptr2 = rdptr_misaligned
ptr3 = wrptr
ptr3
ptr1,2
from APPLICATION

Figure 47

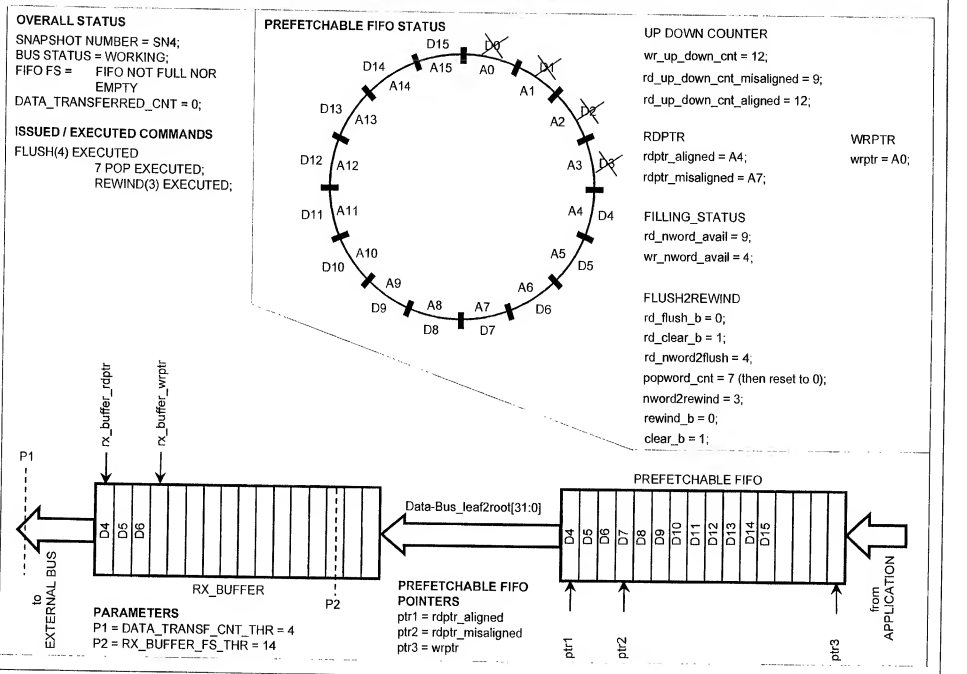

OVERALL STATUS
SNAPSHOT NUMBER = SN4;
BUS STATUS = WORKING;
FIFO FS = FIFO NOT FULL NOR EMPTY
DATA_TRANSFERRED_CNT = 0;
ISSUED / EXECUTED COMMANDS
FLUSH(4) EXECUTED
7 POP EXECUTED;
REWIND(3) EXECUTED;
PREFETCHABLE FIFO STATUS
D15 D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14
A15 A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14
UP DOWN COUNTER
wr_up_down_cnt = 12;
rd_up_down_cnt_misaligned = 9;
rd_up_down_cnt_aligned = 12;
RDPTR
rdptr_aligned = A4;
rdptr_misaligned = A7;
WRPTR
wrptr = A0;
FILLING_STATUS
rd_nword_avail = 9;
wr_nword_avail = 4;
FLUSH2REWIND
rd_flush_b = 0;
rd_clear_b = 1;
rd_nword2flush = 4;
popword_cnt = 7 (then reset to 0);
nword2rewind = 3;
rewind_b = 0;
clear_b = 1;
rx_buffer_rdptr
rx_buffer_wrptr
P1
D4 D5 D6
RX_BUFFER
P2
to EXTERNAL BUS
PARAMETERS
P1 = DATA_TRANSF_CNT_THR = 4
P2 = RX_BUFFER_FS_THR = 14
Data-Bus_leaf2root[31:0]
PREFETCHABLE FIFO
D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15
PREFETCHABLE FIFO POINTERS
ptr1 = rdptr_aligned
ptr2 = rdptr_misaligned
ptr3 = wrptr
ptr1
ptr2
ptr3
from APPLICATION

Figure 48

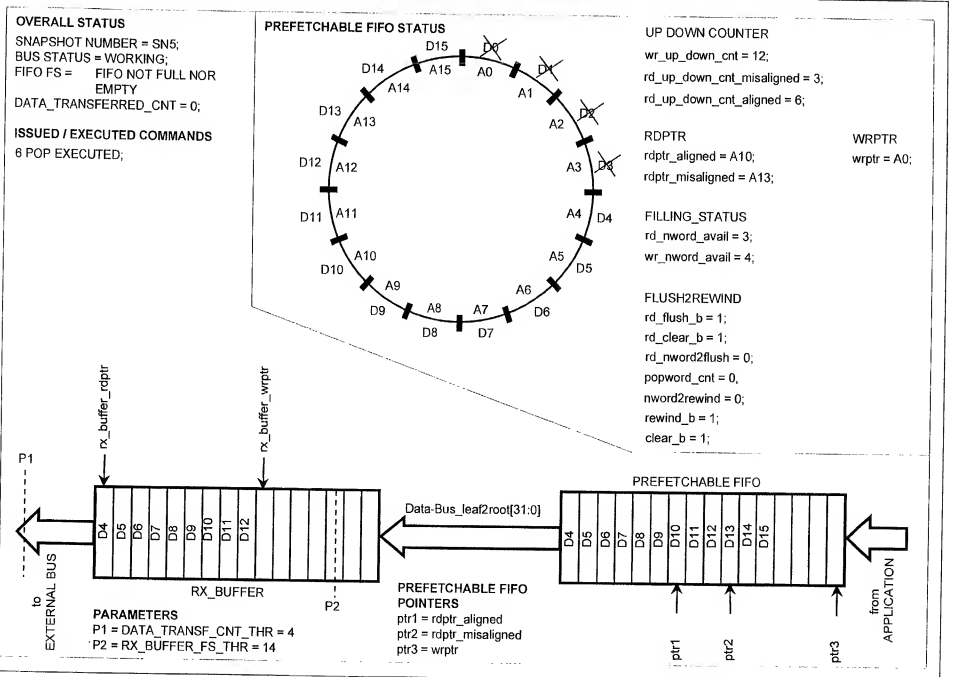

OVERALL STATUS
SNAPSHOT NUMBER = SN5;
BUS STATUS = WORKING;
FIFO FS = FIFO NOT FULL NOR EMPTY
DATA_TRANSFERRED_CNT = 0;
ISSUED / EXECUTED COMMANDS
6 POP EXECUTED;
PREFETCHABLE FIFO STATUS
D15 D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14
A15 A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14
UP DOWN COUNTER
wr_up_down_cnt = 12;
rd_up_down_cnt_misaligned = 3;
rd_up_down_cnt_aligned = 6;
RDPTR
rdptr_aligned = A10;
rdptr_misaligned = A13;
WRPTR
wrptr = A0;
FILLING_STATUS
rd_nword_avail = 3;
wr_nword_avail = 4;
FLUSH2REWIND
rd_flush_b = 1;
rd_clear_b = 1;
rd_nword2flush = 0;
popword_cnt = 0,
nword2rewind = 0;
rewind_b = 1;
clear_b = 1;
rx_buffer_rdptr
rx_buffer_wrptr
P1
D4 D5 D6 D7 D8 D9 D10 D11 D12
RX_BUFFER
P2
to EXTERNAL BUS
PARAMETERS
P1 = DATA_TRANSF_CNT_THR = 4
P2 = RX_BUFFER_FS_THR = 14
Data-Bus_leaf2root[31:0]
PREFETCHABLE FIFO
D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15
PREFETCHABLE FIFO POINTERS
ptr1 = rdptr_aligned
ptr2 = rdptr_misaligned
ptr3 = wrptr
ptr1
ptr2
ptr3
from APPLICATION

Figure 49

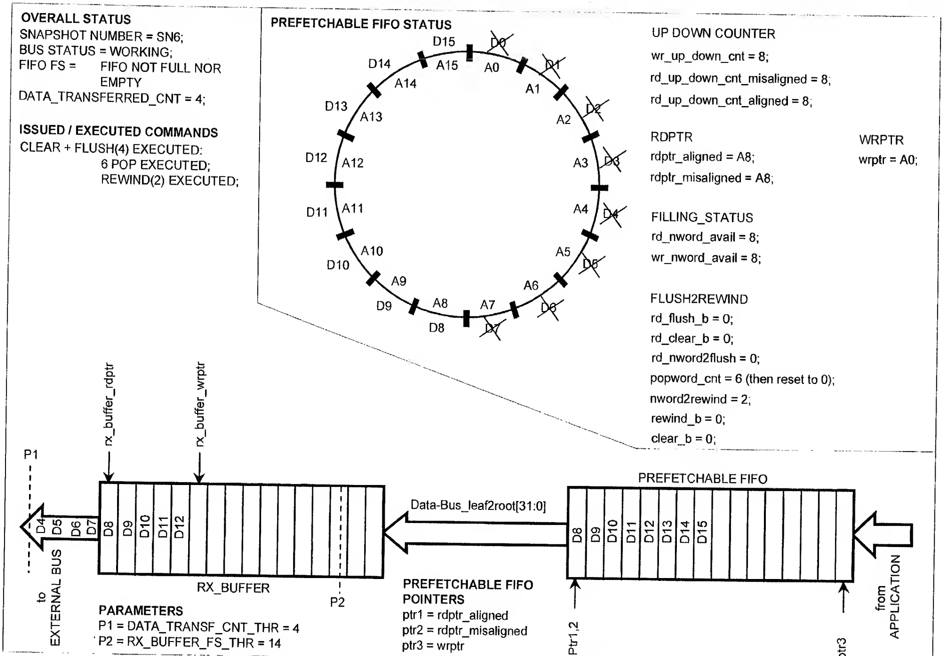

OVERALL STATUS
SNAPSHOT NUMBER = SN6;
BUS STATUS = WORKING;
FIFO FS = FIFO NOT FULL NOR EMPTY
DATA_TRANSFERRED_CNT = 4;
ISSUED / EXECUTED COMMANDS
CLEAR + FLUSH(4) EXECUTED:
6 POP EXECUTED;
REWIND(2) EXECUTED;
PREFETCHABLE FIFO STATUS
D15
D0
A15
A0
D14
D1
A14
A1
D13
A13
A2
D2
D12
A12
A3
D3
D11
A11
A4
D4
D10
A10
A5
D5
A9
A6
D9
A8
A7
D6
D8
D7
UP DOWN COUNTER
wr_up_down_cnt = 8;
rd_up_down_cnt_misaligned = 8;
rd_up_down_cnt_aligned = 8;
RDPTR
rdptr_aligned = A8;
rdptr_misaligned = A8;
WRPTR
wrptr = A0;
FILLING_STATUS
rd_nword_avail = 8;
wr_nword_avail = 8;
FLUSH2REWIND
rd_flush_b = 0;
rd_clear_b = 0;
rd_nword2flush = 0;
popword_cnt = 6 (then reset to 0);
nword2rewind = 2;
rewind_b = 0;
clear_b = 0;
P1
rx_buffer_rdptr
rx_buffer_wrptr
PREFETCHABLE FIFO
D4
D5
D6
D7
D8
D9
D10
D11
D12
Data-Bus_leaf2root[31:0]
D8
D9
D10
D11
D12
D13
D14
D15
to
EXTERNAL BUS
RX_BUFFER
P2
PREFETCHABLE FIFO
POINTERS
ptr1 = rdptr_aligned
ptr2 = rdptr_misaligned
ptr3 = wrptr
Ptr1,2
ptr3
from
APPLICATION
PARAMETERS
P1 = DATA_TRANSF_CNT_THR = 4
P2 = RX_BUFFER_FS_THR = 14

Figure 50

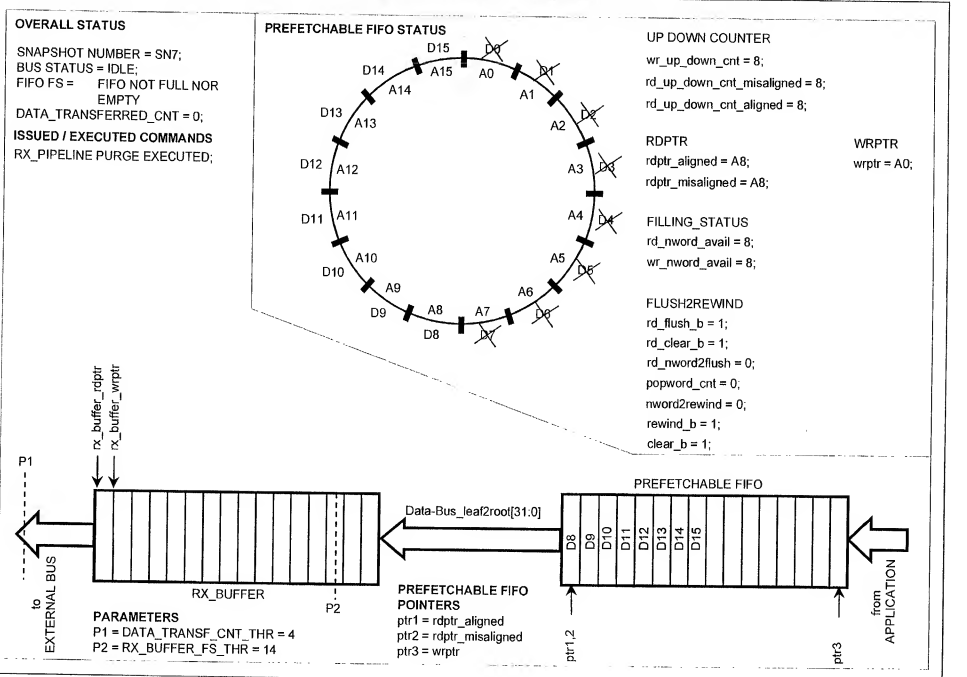

OVERALL STATUS
SNAPSHOT NUMBER = SN7;
BUS STATUS = IDLE;
FIFO FS = FIFO NOT FULL NOR EMPTY
DATA_TRANSFERRED_CNT = 0;
ISSUED / EXECUTED COMMANDS
RX_PIPELINE PURGE EXECUTED;
PREFETCHABLE FIFO STATUS
D15 D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14
A15 A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14
UP DOWN COUNTER
wr_up_down_cnt = 8;
rd_up_down_cnt_misaligned = 8;
rd_up_down_cnt_aligned = 8;
RDPTR
rdptr_aligned = A8;
rdptr_misaligned = A8;
WRPTR
wrptr = A0;
FILLING_STATUS
rd_nword_avail = 8;
wr_nword_avail = 8;
FLUSH2REWIND
rd_flush_b = 1;
rd_clear_b = 1;
rd_nword2flush = 0;
popword_cnt = 0;
nword2rewind = 0;
rewind_b = 1;
clear_b = 1;
rx_buffer_rdptr
rx_buffer_wrptr
P1
PREFETCHABLE FIFO
Data-Bus_leaf2root[31:0]
D8 D9 D10 D11 D12 D13 D14 D15
to EXTERNAL BUS
RX_BUFFER
P2
from APPLICATION
PARAMETERS
P1 = DATA_TRANSF_CNT_THR = 4
P2 = RX_BUFFER_FS_THR = 14
PREFETCHABLE FIFO POINTERS
ptr1 = rdptr_aligned
ptr2 = rdptr_misaligned
ptr3 = wrptr
ptr1,2
ptr3

Figure 51

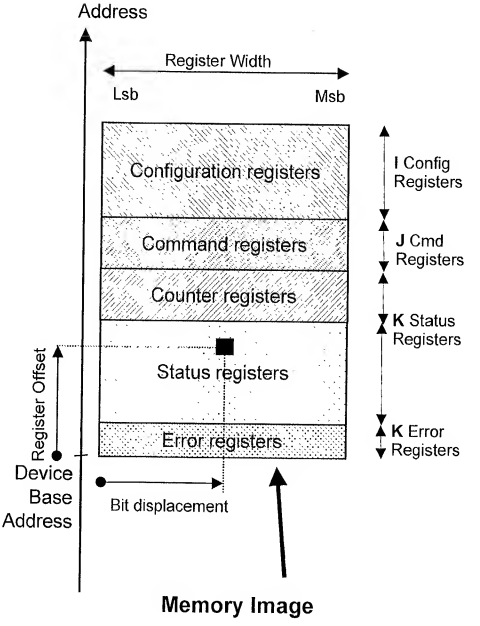

Address
Register Width
Lsb
Msb
Configuration registers
Command registers
Counter registers
Status registers
Error registers
I Config Registers
J Cmd Registers
K Status Registers
K Error Registers
Register Offset
Device Base Address
Bit displacement
Memory Image

Figure 52

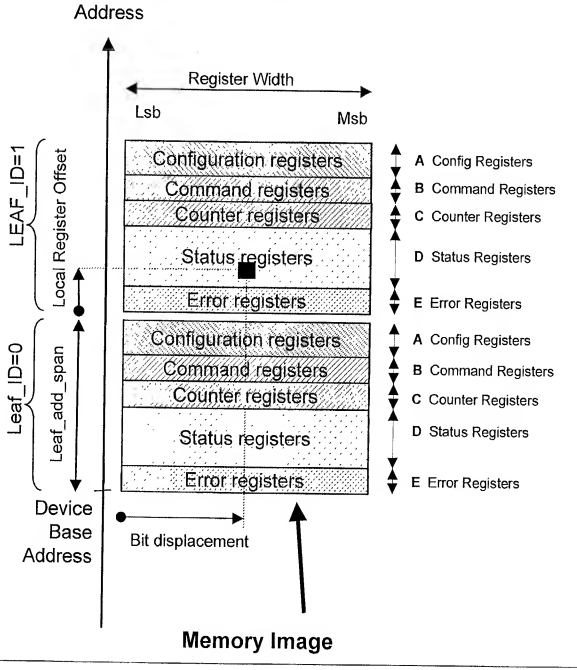

Address
Register Width
Lsb
Msb
LEAF_ID=1
Local Register Offset
Configuration registers
Command registers
Counter registers
Status registers
Error registers
A Config Registers
B Command Registers
C Counter Registers
D Status Registers
E Error Registers
Leaf_ID=0
Leaf_add_span
Configuration registers
Command registers
Counter registers
Status registers
Error registers
A Config Registers
B Command Registers
C Counter Registers
D Status Registers
E Error Registers
Device Base Address
Bit displacement
Memory Image

Figure 53

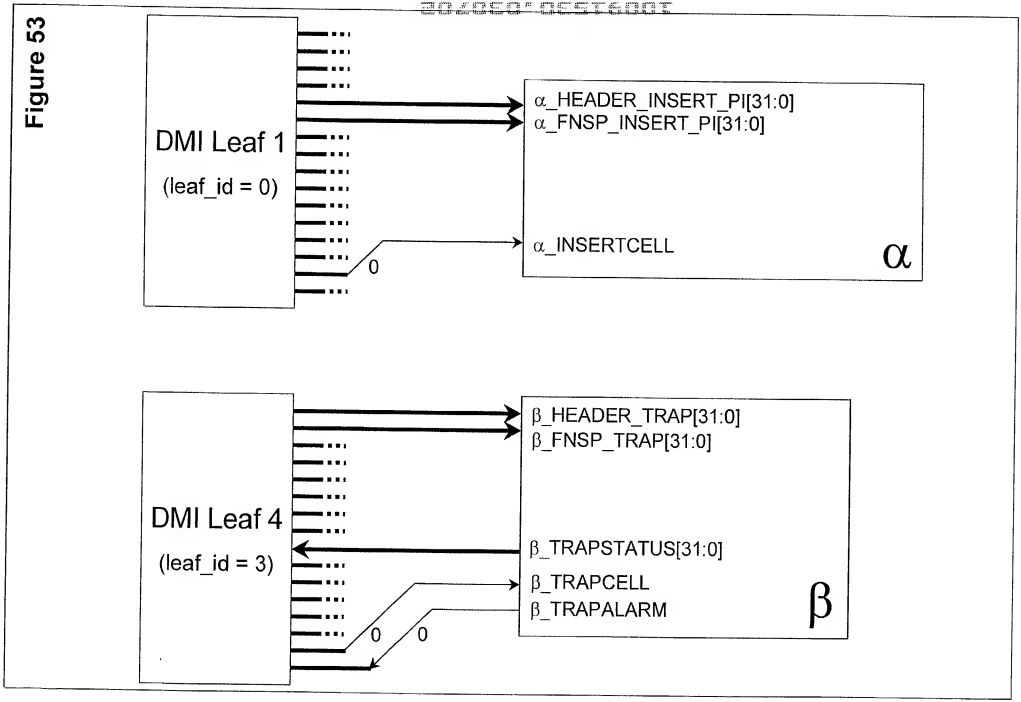

DMI Leaf 1
(leaf_id = 0)
α_HEADER_INSERT_PI[31:0]
α_FNSP_INSERT_PI[31:0]
α_INSERTCELL
0
α
DMI Leaf 4
(leaf_id = 3)
β_HEADER_TRAP[31:0]
β_FNSP_TRAP[31:0]
β_TRAPSTATUS[31:0]
β_TRAPCELL
β_TRAPALARM
0
0
β

Figure 54

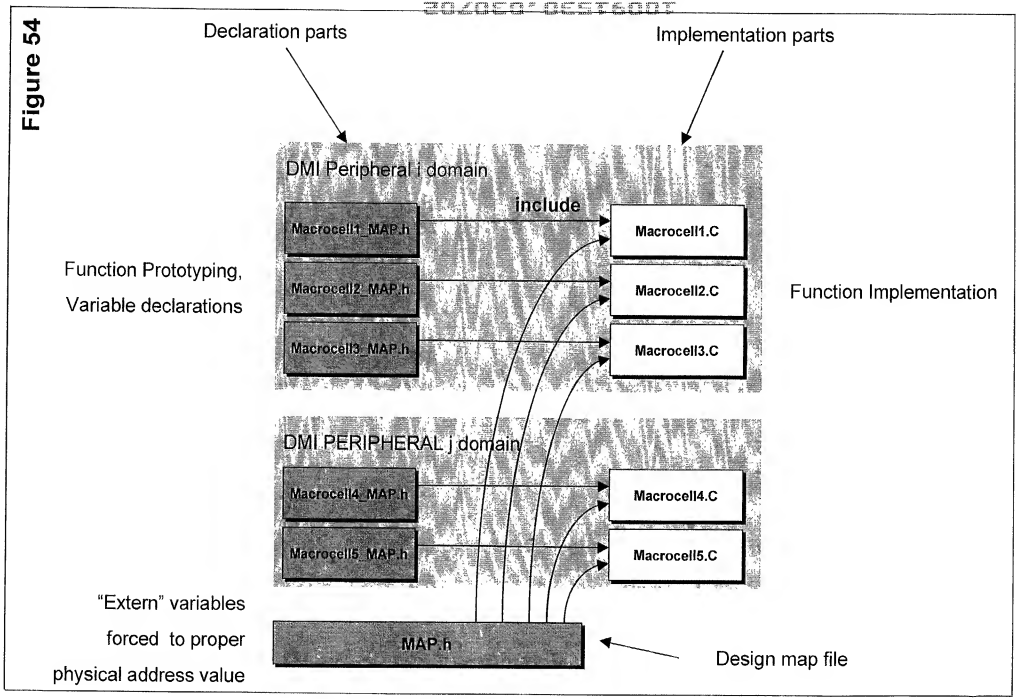
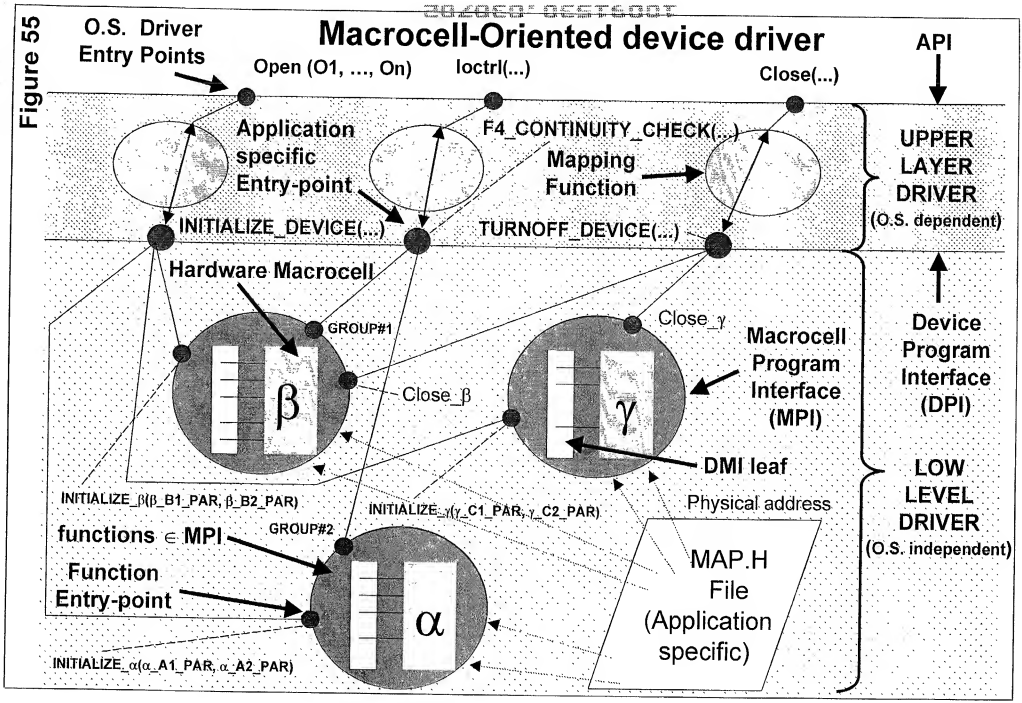

Declaration parts
Implementation parts
DMI Peripheral i domain
include
Macrocell1_MAP.h
Macrocell2_MAP.h
Macrocell3_MAP.h
Macrocell1.C
Macrocell2.C
Macrocell3.C
Function Prototyping,
Variable declarations
Function Implementation
DMI PERIPHERAL j domain
Macrocell4_MAP.h
Macrocell5_MAP.h
Macrocell4.C
Macrocell5.C
"Extern" variables
forced to proper
physical address value
MAP.h
Design map file

Figure 55


Macrocell-Oriented device driver
O.S. Driver Entry Points
Open (O1, ..., On)
Ioctrl(...)
Close(...)
API
Application specific Entry-point
F4_CONTINUITY_CHECK(...)
Mapping Function
UPPER LAYER DRIVER (O.S. dependent)
INITIALIZE_DEVICE(...)
TURNOFF_DEVICE(...)
Hardware Macrocell
GROUP#1
Close_γ
Macrocell Program Interface (MPI)
Device Program Interface (DPI)
β
Close_β
γ
DMI leaf
LOW LEVEL DRIVER (O.S. independent)
INITIALIZE_β(β_B1_PAR, β_B2_PAR)
INITIALIZE_γ(γ_C1_PAR, γ_C2_PAR)
Physical address
functions ∈ MPI
GROUP#2
MAP.H File (Application specific)
Function Entry-point
α
INITIALIZE_α(α_A1_PAR, α_A2_PAR)